High Performance Integrated Circuits and Systems Design Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Departament d'Enginyeria Electrònica

Edifici C4
Campus Nord
c/. Jordi Girona, 1-3
ES 08034 Barcelona
Tel. +34 93 4016748
Fax +34 93 4016756

Internal Research Report:

Set-up of Assura RCX-HF tools for the AMS S35 process. Configuration files and usage guide.[1]

Revision history:

Rev 1.0 October 31st 2008
Xavier Aragones, created.

Index

[1] This work has been supported by projects TEC2005-06784-C02-01 and -C02-02 of the Spanish MEC and FEDER funds

I. Files configured to enable substrate extraction with AssuraRCX-HF for AMS S35 technology

Note: The objective of this section is to document the creation or modification of files to enable substrate parasitics extraction with AssuraRCX-HF for the AMS S35 technology. For a description of the objective and the contents of these files, please refer to the document "Files required for substrate parasitics extraction with Assura RCX-HF".

I.1 - Running snaTCT to create the `SCtechnology.AMS` file

We use SCtechnology.AMS file which is an encrypted file generated by AMS (30th April 2003). It was generated from the substrate profiles of the C35 process. By request of UPC, N-well capacitances were given the values calculated from the Process Parameters (CJNW=0.03699 fF/μm^2, CJSWNW=0.2989 fF/μm^2), assuming a 3 V bias voltage.

In that technology file, the following regions and cross-sections are defined:

`default` Region:
Cross-sections: `default ntub contact channel sd`

`ntub` Region:
Cross-sections: `default contact channel sd`

For more information about running snaTCT, please refer to the "`RCX-HF Substrate Technology Characterization Manual`". For a recent example of creating a SCtechnology file, please refer to the reports written for the CTH process.

I.2 - Modifications on the `compare.rul` file

This section describes the modification that needs to be done to the `compare.rul` file, located under `<install_dir>/assura/s35d4/s35d4m5/`. This modification is necessary to filter devices added by Assura-RCX_HF during the extraction process, but not present in the layout, in order to avoid LVS mismatches.

If you cannot modify the file in the installation directory, please copy the complete folder to your working directory, and specify the new file location when running LVS, in the LVS form.

The modification consists of introducing the following lines inside the
```
avCompareRules(
  layout(
```
section of the `compare.rul` file.

Lines to be added:

```
    filterDevice( "short") ; currently not used
    filterDevice( "short_ptap") ; device for ptaps in default regions
    filterDevice( "short_ntap") ; device for ntaps in Nwell regions
    filterDevice( "TIE") ; currently not used
    filterDevice( "TIN") ; currently not used
```

I.3 - Modifications on the `extract.rul` file

This section describes the modifications that need to be done to the `extract.rul` file, located under located under `<install_dir>/assura/s35d4/s35d4m5/`

If you cannot modify the file in the installation directory, please copy the complete folder to your working directory, and specify the new file location when running LVS, in the LVS form.

1.3.0.- General description:

To better understand the modifications to this file, it is worth briefly describing the fundamental actions that are performed in the `extract.rul` file.

First, the basic layers (present in the layout) are identified and assigned labels.

Second, from logical functions applied to these basic layers, new layers are defined to identify basic *technological regions*. For example, this includes the identification of diffusions, wells, etc (`net_pdiff, net_ndiff, net_psub, net_nwell...`).

Third, from the above defined layers, new layers are defined to identify basic *functional regions*. For example, this includes the identification of drains, sources, gates, taps, etc. (`net_psd, net_nsd, net_subtap, net_welltap...`).

Fourth, from the above defined layers, new layers are defined to identify devices (`NMOS_device, PMOS_device, RNWELL_device )`

Fifth, electrical connections between layers are defined (note that the layers identified up to now were just geometrical layers). For example, if a contact is defined to connect metal1 and the source/drain, tap and/or diffusion (`via( net_pdiffcon net_met1 net_psd net_subtap ) )`

Sixth, devices extracted from the layers before. This includes identify the layers connected to each terminal, assign a model, obtain parameters, etc,

Last, a `saveInterconnect` sentence allows saving those devices or layers that we want to appear in the `av_extracted` view.

I.3.1.- Devices:

It is necessary to save the recognition shapes of all the devices that will be access ports to the substrate mesh (i.e., devices that will interact with the substrate model). These should at least include the devices listed in the `snaGeneration` (and `snaSelection`) sections of the `SCparameters.cds` file (see section I.4.1).

For the S35 process this has not been necessary since the basic devices included in the `SCparameters.cds` file already had the recognition shapes saved.

Example:

```
extractMOS( "NMOS4" NMOS_device
      (net_poly1 "G") (net_nsd   "S" "D") (net_psub "B")
      targetLayer(net_poly1)
      cellView("nmos4 auLvs PRIMLIB") spiceModel("MODN")
      )
saveRecognition( NMOS_device "device" )
```

If new devices are included to the file `SCparameters.cds`, the `extract.rul` file should be revised to check that a `saveRecognition` sentence is present after extraction of each device.

Notes:
1.- According to the layer definition of layers in the extraction rules file, recognition shapes of MOSFET transistors correspond to their gates. Thus this will be the area of the access ports for MOSFETs.
2.- Extraction of RF devices, which include a substrate mesh in the models, should have a particular treatment.

1.3.2.- Substrate

When extracting substrate parasitics, it is usually desired to model the parasitic transmission path between devices in noisy and sensitive domains. Usually these devices will belong to different power-supply domains (eg., digital and analog). Therefore, it is assumed that different GND nodes will be connected to the substrate. By default, the substrate (`net_psub`) is a single region covering all the chip except the N-wells, therefore all substrate nodes will appear shorted in the extraction (`net_psub` is shorted to `net_subtap`). This has proved to be a problem when different GND nodes are used, regardless of the extraction (or not) of the substrate. We have tested several methods to circumvent this problem, but the only one that has worked without any problems is to use a definition layer to specify the different power-supply domains.

AMS includes a definition layer called SUBDEF, and by default this layer is used to define the geometry of `net_psub`. Therefore, whenever different voltage domains are present in a layout, SUBDEF layer should be used to distinguish between them. SUBDEF polygons should cover all the area of each domain, while they do not overlap with SUBDEF polygons of other domains. This will guarantee that the respective substrates are connected to the different GND nodes, while a single GND node is defined within each region.

We have added a sentence to define `net-psub` only as the region overlapping SUBDEF (this avoids the appearance of an extra "net_sub" region between two "net_psub" domains)

```
net_psub    = geomAnd(net_psub1 SUBDEF)
```

Also we have included a sentence in the `saveInterconnect` section to allow visualizing the different substrate regions, in the `av_extracted` view.

```
(net_psub "psub_esd") ; added by XA
```

It is important to note that, with these modifications, it is mandatory that any device in the layout is covered by some `SUBDEF` rectangle, even if there is a single voltage domain. Otherwise, the LVS will fail.

1.3.3.- N-well:

A special layer `sna_ntub_region` is defined to identify the N-well regions for which the junction capacitance will be extracted. These are all the N-well areas except those that already belong to some device (N-well resistor, BJT...) for which the junction is already included in the device model. Those areas are identified as `no_sna_ntub_region` (in the `extract.rul` version that we used, these layers were already defined).

```
sna_ntub_region = geomAndNot( net_nwell no_sna_ntub_region)
```

The new `sna_ntub_region` layer is electrically connected to the node of the N-well:

```
geomStamp( sna_ntub_region net_nwell)
```

Recognition shape of the well region has been added, thus the substrate tool will be able to distinguish between the regions with different profiles and wells. Recognition shape is saved to the `NTUB` layer (Apparently, this sentence is not necessary since the first line in the `SaveInterconnect` command also saves the recognition shape).

```
saveDerived( sna_ntub_tap ("NTUB" "drawing") ext_view)
```

A sentence is included in the `saveInterconnect` section to allow visualizing the different well regions in the `av_extracted` view.

```
(sna_ntub_region "NTUB") ;;SubstrateStorm
```

Last, the sentences that allowed extracting the N-well junction diodes have been eliminated (commented), in order to avoid duplication with the junction capacitances that are already extracted together with the substrate.

```
;;extractDIODE( "NWD" NWD_device
      ...
;;extractDIODE( "P_NWD" P_NWD_device
      ...
```

It is important to note that, since the above diode extraction has been eliminated, no junction parasitics will be extracted if the substrate is not extracted. In order to avoid these kind of problems, it is recommended that a switch is defined and the different sentences valid for substrate extraction are grouped under a switch condition.

1.3.4.- Ptaps and Ntaps:

Biasing contacts (Ptaps and Ntaps) must be access ports to the substrate mesh. Therefore, they must be extracted.

Special layers `sna_psub_tap` and `sna_ntub_tap` are defined to identify the psub and nwell taps that must be extracted. These are all the taps except those that already belong to some device (N-well resistor, BJT...) for which well connection is already included in the device model. Those areas are identified as `no_sna_psub_tap` and `no_sna_ntub_tap` (in the `extract.rul` version that we used, these layers were already defined).

```
sna_psub_tap = geomAndNot( net_subtap no_sna_psub_tap)
sna_ntub_tap = geomAndNot( net_welltap no_sna_ntub_tap)
```

The `sna_psub_tap` and `sna_ntub_tap` layers are electrically connected to the diffusions:

```
geomStamp( sna_psub_tap net_pdiff )
geomStamp( sna_ntub_tap net_ndiff )
```

Although AssuraRCX-HF manuals recommend extracting the taps as `TIE` (and `TIN`) devices, we have not suceeded to get a single sucessful extraction of the tap connections with these devices (`TIE` are identified as ports of the substrate mesh, and the substrate subcircuit is correctly generated, but the final netlist shows that the electrical connection between the circuit and the substrate subckt is not done properly).

Therefore, we used as an alternative solution the extraction of small resistances as `short_ptap` (and `short_ntap`) devices. This solution has the disadvantage of including an extra resistor in the netlist for each tap (there should appear no extra devices with the `TIE` solution), but at least works, and does not electrically alter the circuit, since the resistance extracted has an extremely low value (10^{-9} Ω).

When the substrate is not extracted, both ends of the `short` device are the same node (GND or V_{dd}). When the substrate is extracted, one end of the `short` ends is connected to GND (or V_{dd}), the other end to the node of the substrate access port.

Recognition shaped for these `short_ptap` and `short_ntap` devices are also saved.

```
    extractDevice( "short_ptap" sna_psub_tap net_pdiff( "PLUS" )
net_psub( "MINUS" )
        cellView( "short_ptap ivpcell substrateLib" ) )
    saveProperty( sna_psub_tap "r" "1e-9" )
    saveRecognition( sna_psub_tap "device" )

    extractDevice( "short_ntap" sna_ntub_tap net_ndiff( "PLUS" )
net_nwell( "MINUS" )
        cellView( "short_ntap ivpcell substrateLib" ) )
    saveProperty( sna_ntub_tap "r" "1e-9" )
    saveRecognition( sna_ntub_tap "device" )
```

Finally, sentences are included in the `saveInterconnect` section to allow visualizing the different taps in the `av_extracted` view.

```
      (sna_psub_tap "psub_tap") ;;SubstrateStorm
      (sna_ntub_tap "ntub_tap") ;;SubstrateStorm
```

I.4 - Description of the `SCparameters.cds` file

Note: The objective of this section is to document the creation of the `SCparameters.cds` file for the AMS S35 process. For a complete description of the contents and syntax of this file, please refer to the Assura Physical Verification User Guide (Product version 3.1.5_OA), Appendix D, as well as the .Assura Physical Verification Developer Guide (Product version 3.1.5_OA), Chapter 10

1.4.1. `snaGeneration` section

```
snaGeneration = '(

 ( "short_ptap" nil (( "MINUS" )( "PLUS" )) "default" "contact"
"Resistive" "device" "Unique" 1e4 -1 0 0 "new" )
 ( "short_ntap" nil (( "MINUS" )( "PLUS" )) "ntub" "contact"
"Resistive" "device" "Unique" 1e4 -1 0 0 "new" )

 ( "nmos4" nil (( "B" )( "D" "S" )( "G" )) "default" "channel"
"Resistive" "device" "Unique" 1e4 -1 0 0 "new" )
 ( "pmos4" nil (( "B" )( "D" "S" )( "G" )) "ntub" "channel"
"Resistive" "device" "Unique" 1e4 -1 0 0 "new" )
```

`short_ptap, short_ntap, nmos4, pmos4` are names of extracted devices, as defined in the `extract.rul` file.

`default, ntub` are names of regions as defined in the `SCtechnology.AMS` file.

`contact, channel`, are names of cross-sections as defined in the `SCtechnology.AMS` file.

`Resistive` connection is chosen since all depletion capacitances at the device-substrate interface are already included in the models of the devices.

`NEW` specifies that a new node is generated to connect the `B` (bulk) terminal of the devices to the substrate. Alternatively, `INLINE` specifies that the ideal node name is preserved. In the case of `short_ptap` or `short_ntap` devices, `NEW` only makes sense when the parasitics of the supply distribution are extracted (IR drop accounted for). If all the supply distribution is a single node, it will short-circuit all the `short_ptap` (or `short_ntap`) access ports, thus it makes no sense to specify different node names for these access ports. In this case, it is better to use the `INLINE` parameter, which will produce a more compact substrate netlist.

For each of the devices defined, the first node in the list is that connected to the substrate mesh (`B` in MOSFETS, `PLUS` in n-diffusion diodes, `MINUS` in p-diffusion diodes, etc). The node names are defined in the `extract.rul` file.

Note that currently, only these four devices will be identified as access ports (interaction between device and substrate parasitics is only accounted for these devices). Other devices should be included in this section.

Notes:

1. In a given layout, all instances of devices included in the `snaGeneration` section will become access ports to the substrate. If it is desired that only a fraction of the instances become access ports, that device definition should be moved to the `snaSelection` section, which allows selecting which of the instances of a particular device will be access port. (see section III.2 of this document)
2. Devices that have two-terminal models, no bulk node, cannot be included in this section. The only way to include them as access ports would be through a capacitive connection to one of the terminals (eg, lower plate of a capacitor), capacitance value defined in the technology file.
3. Some devices such as Poly resistors, which can be laid both on P-substrate and on N-wells, cannot be included in this section. Ideally, the `extract.rul` file should be modified in order to extract different devices depending on the well type beneath the device. Then, they could be identified as substrate ports independently, and assigned a `default` or `ntub` substrate region depending on the case.

1.4.2. `snaRegions` section

```
snaRegions = '(
    ( ( "NTUB"  "net" ) "ntub" ( (3.3 "Vdd") ))
)
```

`NTUB` is the layer to which the recognition shape of the Nwell is saved (see section 1.3.3 of this document).

`ntub` is the name of a region as defined in the `SCtechnology.AMS` file.

`Vdd` is the default node connection to the ntub regions, although this will be overwritten when extracting the substrate. Also, the `3.3` voltage is not important given that junction capacitances have fixed (voltage-independent) values.

1.4.3. `snaLayersAndPurposes` section

```
snaLayersAndPurposes = '(
  ( "SCR"   "drawing"  )  ; this LPP is used to display REGIONS
  ( "SCAP"  "drawing"  )  ; this LPP is used to display ACCESS PORTS
  ( "SCAP"  "drawing1" )  ; this LPP is used to HIGHLIGHT ACCESS PORTS
  ( "SCSD"  "drawing"  )  ; this LPP is used to display SURFACE NOISE
DISTRIBUTION for LEVEL 0 ( low noise )
  ( "SCSD"  "drawing1" )  ; this LPP is used to display SURFACE NOISE
DISTRIBUTION for LEVEL 1
  ( "SCSD"  "drawing2" )  ; this LPP is used to display SURFACE NOISE
DISTRIBUTION for LEVEL 2
  ( "SCSD"  "drawing3" )  ; this LPP is used to display SURFACE NOISE
DISTRIBUTION for LEVEL 3
  ( "SCSD"  "drawing4" )  ; this LPP is used to display SURFACE NOISE
DISTRIBUTION for LEVEL 4
  ( "SCSD"  "drawing5" )  ; this LPP is used to display SURFACE NOISE
DISTRIBUTION for LEVEL 5
  ( "SCSD"  "drawing6" )  ; this LPP is used to display SURFACE NOISE
DISTRIBUTION for LEVEL 6
  ( "SCSD"  "drawing7" )  ; this LPP is used to display SURFACE NOISE
DISTRIBUTION for LEVEL 7
  ( "SCSD"  "drawing8" )  ; this LPP is used to display SURFACE NOISE
DISTRIBUTION for LEVEL 8
  ( "SCSD"  "drawing9" )  ; this LPP is used to display SURFACE NOISE
DISTRIBUTION for LEVEL 9 ( high noise )
  ( "SCPP"  "drawing"  )  ; this LPP is used to display PERTURBING
PATH at substrate SURFACE
  ( "SCPP"  "drawing1" )  ; this LPP is used to display PERTURBING
PATH in substrate DEPTH
  ( "SCPP"  "label"    )  ; this LPP is used to display the NOISE
level VALUES
  ( "SCSG"  "drawing"  )  ; this LPP is used to display substrate
surface MESH
  ( "y0" "drawing"  )  ; this LPP is used to display the MACRO port
MASK
)
```

`SCR`, `SCAP`, `SCSD drawingX...` are existing layers defined on purpose to display some graphical representations related with substrate extraction. If these special layers had not been defined, other existing layers should be chosen according exclusively to their color and texture suitability (ex: for the surface map distribution described in section III.3, it is desired that colors follow a uniform degradation).

II. Cadence configuration to run AssuraRCX-HF substrate extraction

Note: Part of these configuration steps are described in the Assura Physical Verification Developer Guide (Product version 3.1.5_OA), Chapter 10. It is assumed that configuration of Cadence and other Assura tools are already properly performed.

1.- In your `.cshrc` or similar, define the path to the SNA[2] module directory and add the executables to your path variable:

```
setenv SUBSTRATESTORMHOME $ASSURAHOME
setenv PATH $SUBSTRATESTORMHOME/bin:$PATH
```

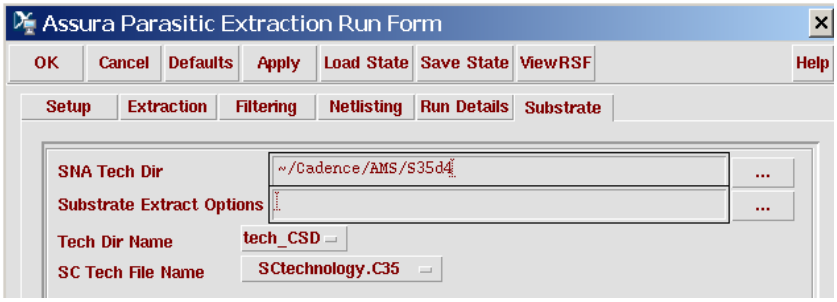
(The SNA application module is found in the Assura software release at `$ASSURAHOME/tools/sna`.)

2.- (This step is not described in the Cadence documentation). A SNA tech directory called `tech_CSD` must exist, containing the SNA tech files `SCtechnology.AMS` and `SCparameters.cds` files. This directory should be part of the design kit installation tree. If substrate extraction capability is not released as part of the design kit, create the SNA tech directory in your local working directory.

3.- In your `.cshrc` or similar, define the path to the SNA tech directory

```
setenv SUBSTRATESTORMSITE <path>
```

Note: This step is necessary to include the *default* path to the SNA tech directory. For substrate extraction this step is not necessary, since this directory can be manually entered in the "RCX Run..." form. Nevertheless, the "Substrate AC Analysis..." feature requests the `SUBSTRATESTORMSITE` variable to be defined.



4.- Edit your `.cdsinit` file (located in your home directory or current working directory) to load the SNA initialization file. This `.ini` file will load the SNA context for use with Assura RF.

```
load("$SUBSTRATESTORMHOME/tools/sna/bin/32bit/etc/context/5.1.0
/sna.ini")
```

Note: According to our tests this step is not necessary

5.- Edit your `cds.lib` or `lib.def` (OA only) file to add the `substrateLib` library:

```
DEFINE substrateLib
$SUBSTRATESTORMHOME/share/sna/etc/cdslib/substrateLib
```

This library contains several devices which are used during the substrate extraction process.

6.- (This step is not described in the Cadence documentation). From Cadence, Library Manager, open the `substrateLib` library. Copy the `short` cell to a new `short_ptap` and `short_ntap` cells in the same library. If you don't have write permissions on the `substrateLib` library, copy it to a local directory and repeat step 5 pointing to your local `substrateLib` copy.

[2] Substrate Noise Analysis

Note: `short_ptap` is the device extracted for substrate ptaps. A `short_ntap` device is necessary for ntaps, which have a different substrate profile (see section I.3.3). Other devices will be needed for other wells.

7.- (This step is not described in the Cadence documentation). A library including parasitic devices (`presistor, pcapacitor`, etc. must be available). This can be the `PRIMLIB` library, or the `analogLib`. In this last case the library path must be explicitly defined in your `cds.lib` file.

III. Usage guide

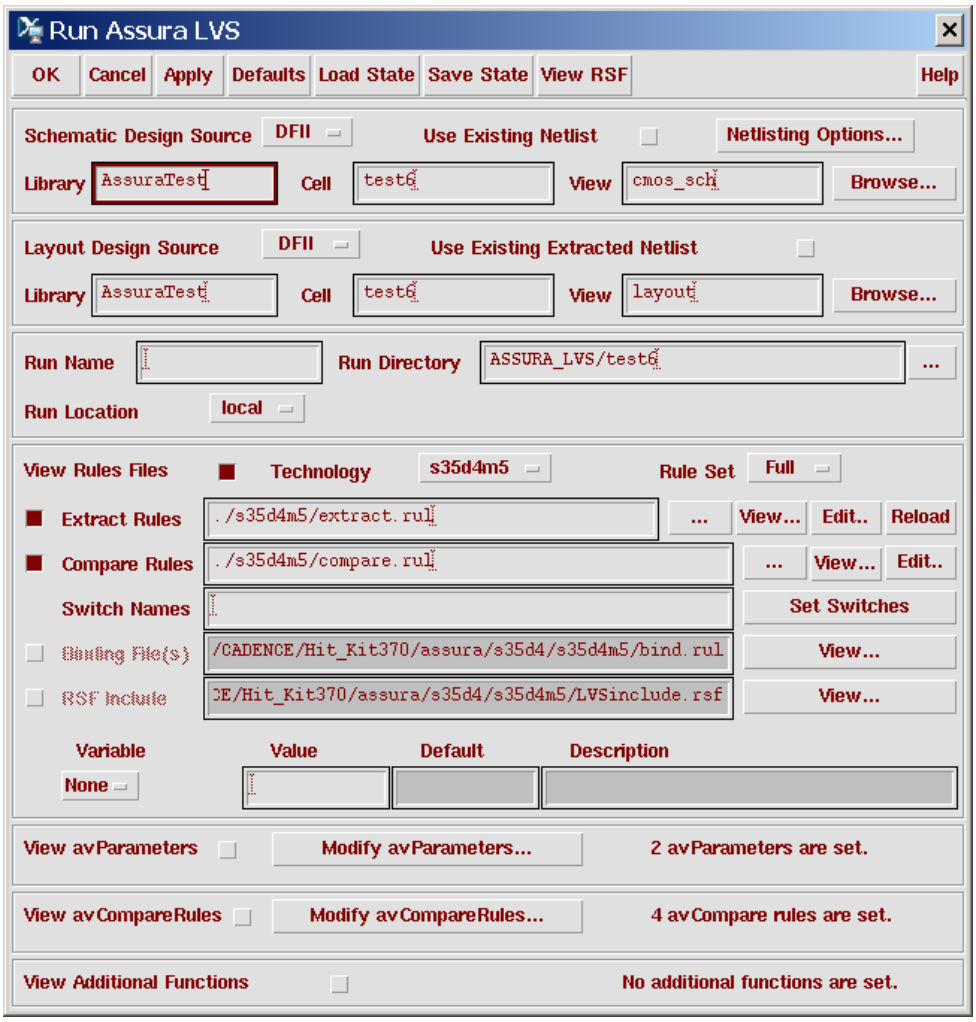
III.1 - Running LVS

III.1.1.- Before running LVS: Important note on multiple supply voltages.

Substrate extraction is often desired for layout situations in which there are at least two different power-supply domains (digital or noisy, analog or quiet). The substrate will be connected to the corresponding supply voltage within each domain. Typically, standard cells have their supply voltages defined as global nodes gnd! and vdd!. This should be modified, at least, for the sensitive circuitry, with quiet supply voltage externally accessible both in the layout and schematic. In our examples, we have also modified the names of the digital supply voltages (no global nodes are used) and the symbols (in order to make the supply nodes accessible). It is not clear how the use of global nodes will affect the extraction.

III.1.2.- LVS run

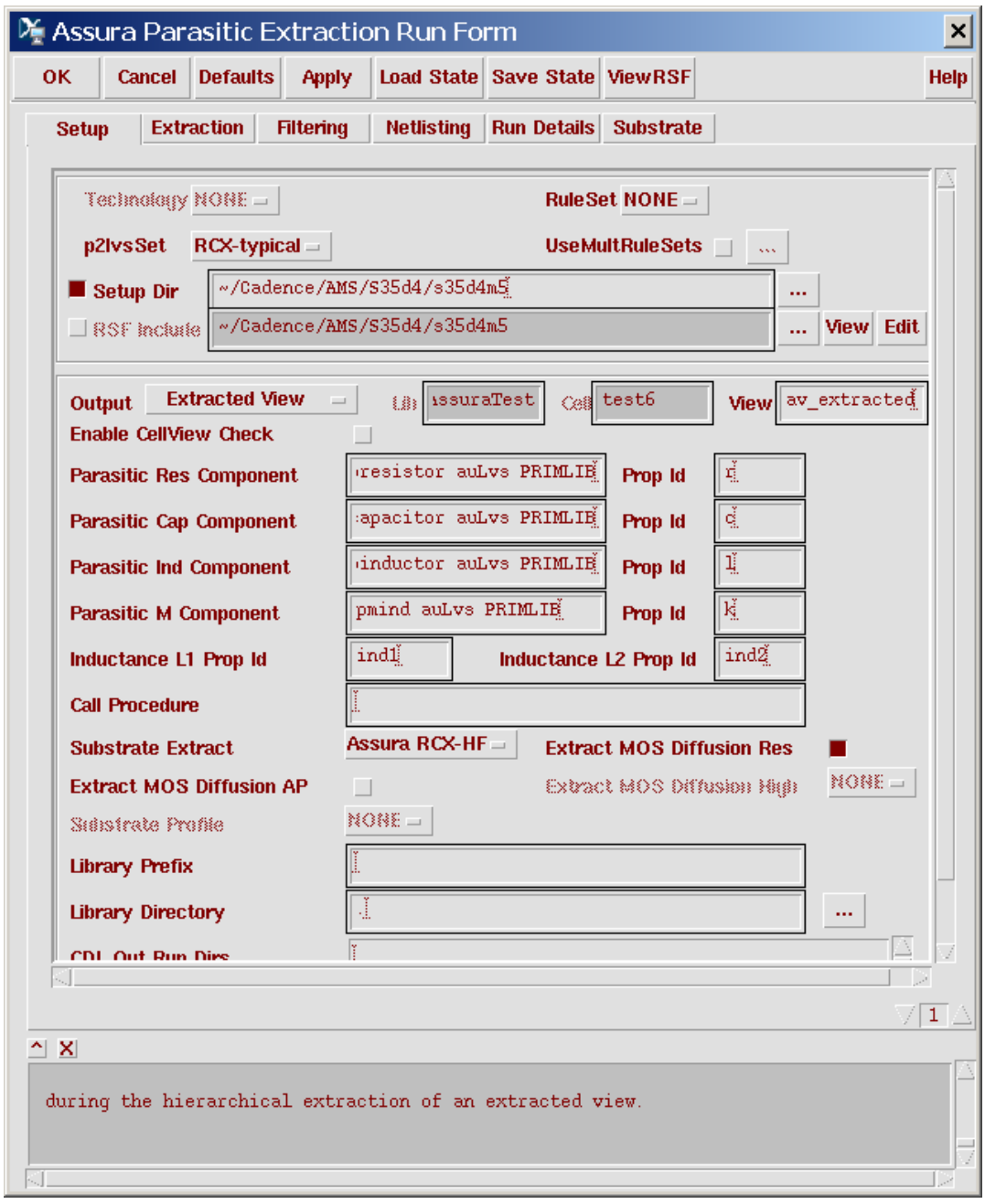
LVS must be run and passed free of errors before running AssuraRCX. It is important that the `extract.rul` and `compare.rul` files contain the modifications Described in sections 1.2 and 1.3. If these modified files are not contained in the PDK installation tree, you can create them and store them in a local directory. In this case, it is important that you specify the paths to these files in the "Run Assura LVS" form, and click on the boxes to specify that you will use these local files and not those of the PDK.



III.2 - Running AssuraRCX

For a complete description of the AssuraRCX tools and the associated forms and options, please refer to the Assura Physical Verification User Guide (Product version 3.1.5_OA).

III.2.1.- Setup tab

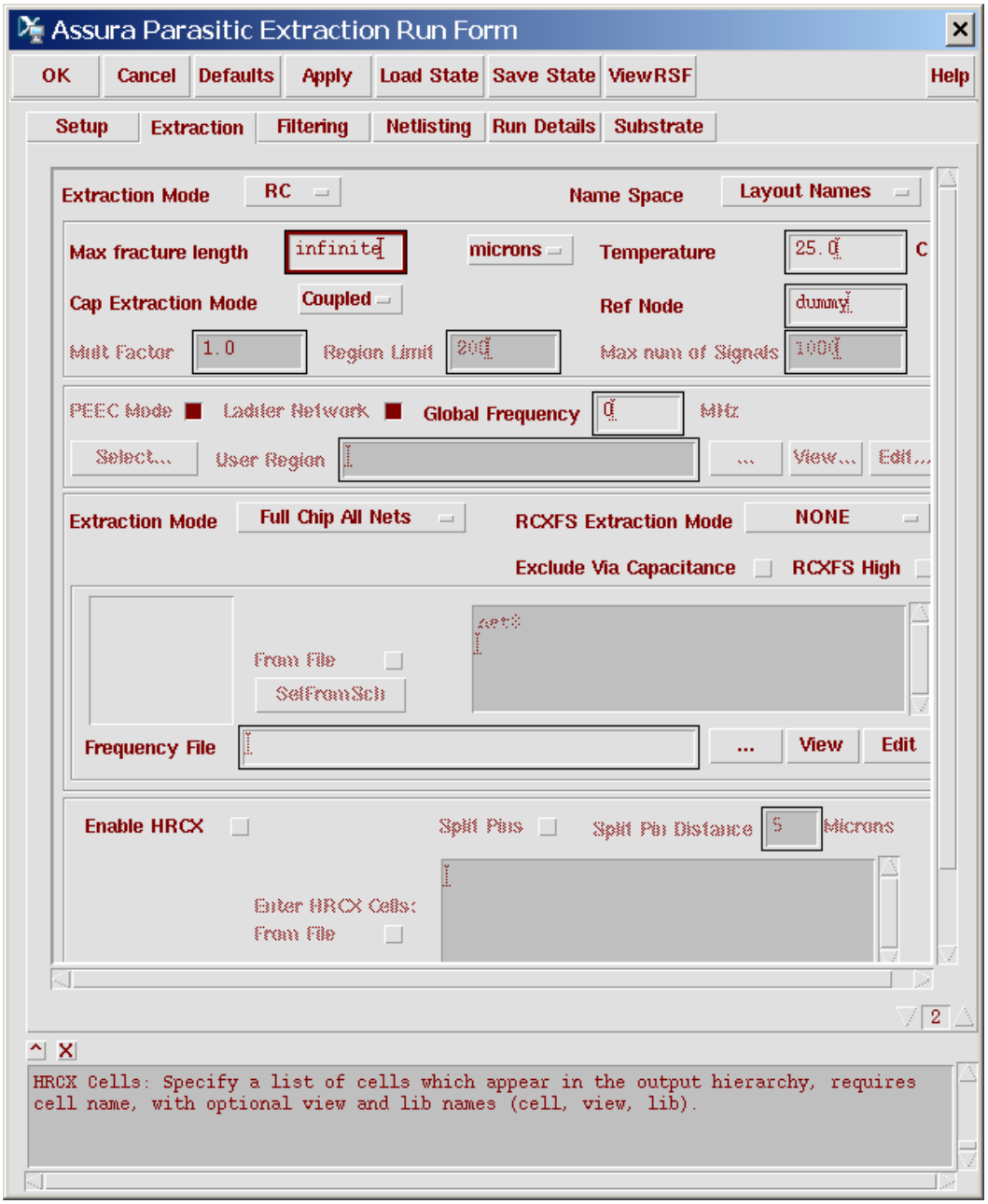


Setup Dir. This should point to the directory where the `RCXspiceINIT` file is found or, if different p2lvsSet exist, to its parent directory (this is the case of AMS S35 technology)

Parasitic XXX Component. See step 7 in section II.

Substrate Extract: Select AssuraRCX-HF to enable substrate extraction

III.2.2.- Extraction tab

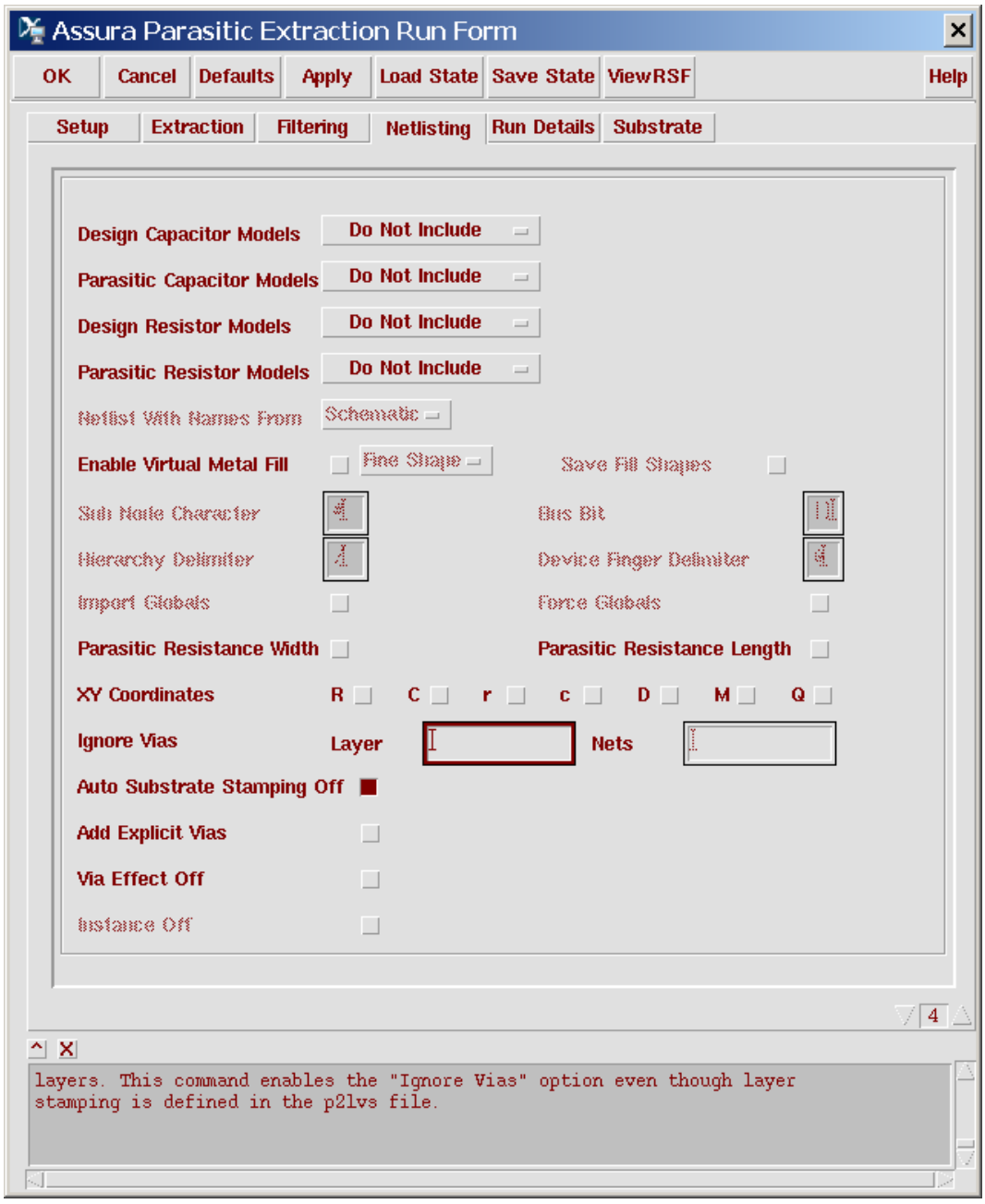


Extraction Mode: There are some restrictions concerning the extraction of interconnects inductances together with the substrate. Please refer to the Assura RCX manual.

Name Space: Layout Names recommended

Ref Node: See example in Appendix A.2

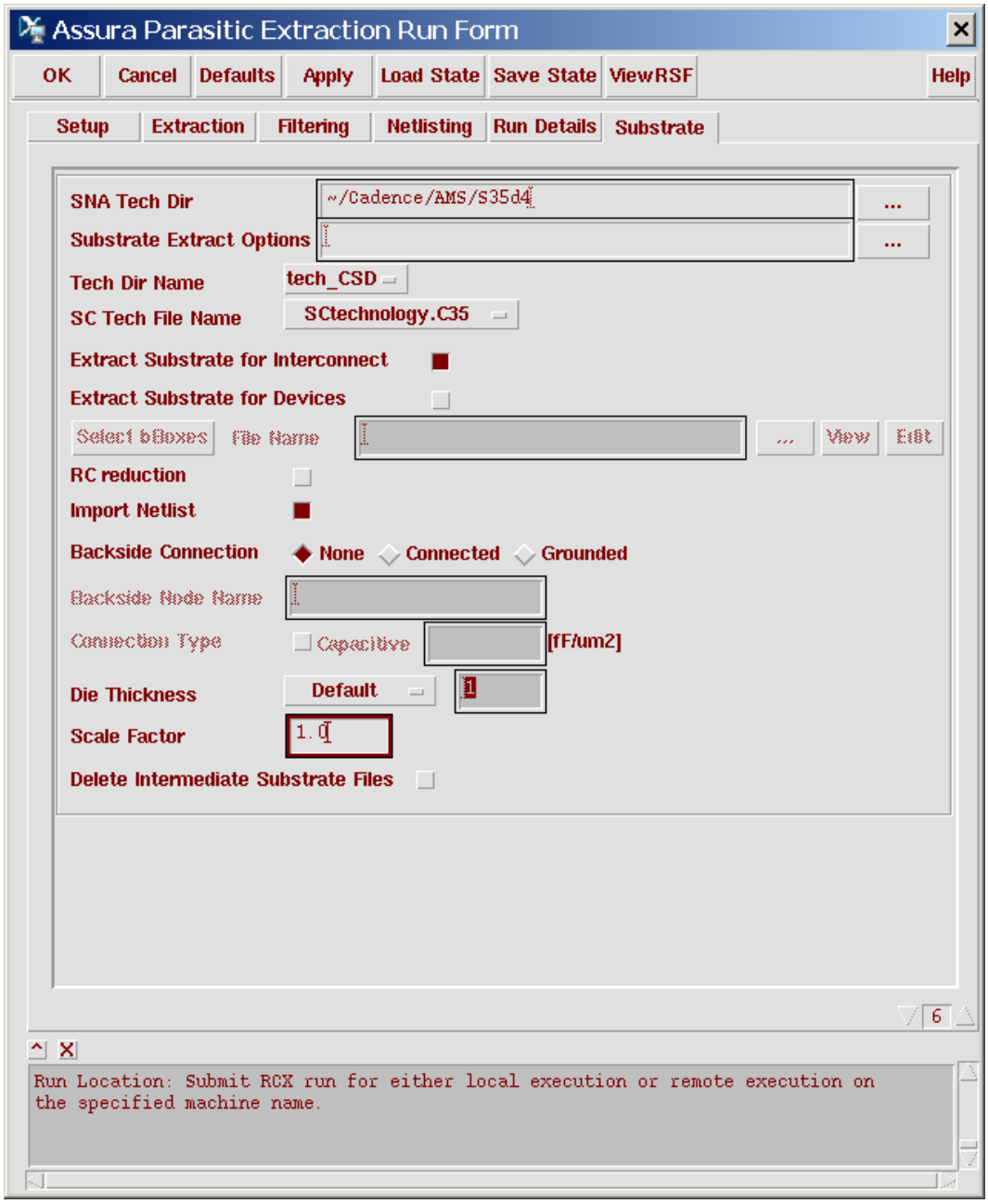
III.2.3.- Netlisting tab



Parasitic Capacitor Models, Parasitic Resistor Models: Do not include these models, these lets Assura use the default models.

Particularly, if Parasitic Capacitor Models is set to "Include Models", then some extracted parasitic capacitors will be extracted as `snacapacitor` and `cmodel` devices. The simulator will complain that it cannot find the definition of these models, and simulation will fail.

III.2.4.- Substrate tab



SNA Tech Dir: Should point to the parent of the `tech_CSD` directory (see step 2 in section II)

Extract Substrate for Interconnect: All instances of the devices listed in the `snaGeneration` section of the `SCparameters.cds` will become access ports.

Extract Substrate for Devices: Only the selected instances of the devices listed in the `snaSelection` section of the `SCparameters.cds` will become access ports. Instances are selected by using bBoxes.

RC reduction: If this option is switched off, the substrate parasitics subscircuit consists of the complete mesh of parasitic resistances and capacitances (of well junctions). The number of components in this subcircuit can range from thousands (for the simplest cells) to hundreds of

thousands. With the RC reduction option switched on, a mathematical postprocessing is done to the subcircuit, which reduces the number of components several orders of magnitude. The resistances and capacitances in the reduced subscircuit no longer have a physical meaning (negative resistances or capacitances may be found), but it is presumed that the overall behavior of the circuit at the frequency of interest is still the same (it is adviceable to verify this assumption with some small circuit).

Import netlist: With this option on, the complete substrate model is listed in the extracted netlist. With this option off, the substrate model is included as a subcircuit call in the extracted netlist.

After running AssuraRCX, an `av_extracted` view is created. A `.sna` directory is created inside the library directory, which contains the substrate extraction data, including a `substrate.subckt` file which contains the substrate extracted model in spice format.

The `av_extracted` view now contains the extracted layout including interconnect parasitics and the substrate parasitics. The extracted view is now normally used: you can now define a simulation set-up, enter the Analog Design Environment and create a circuit netlist. The netlist now contains all the extracted parasitics, including the substrate. A circuit simulation will now reflect possible node interactions through the substrate.

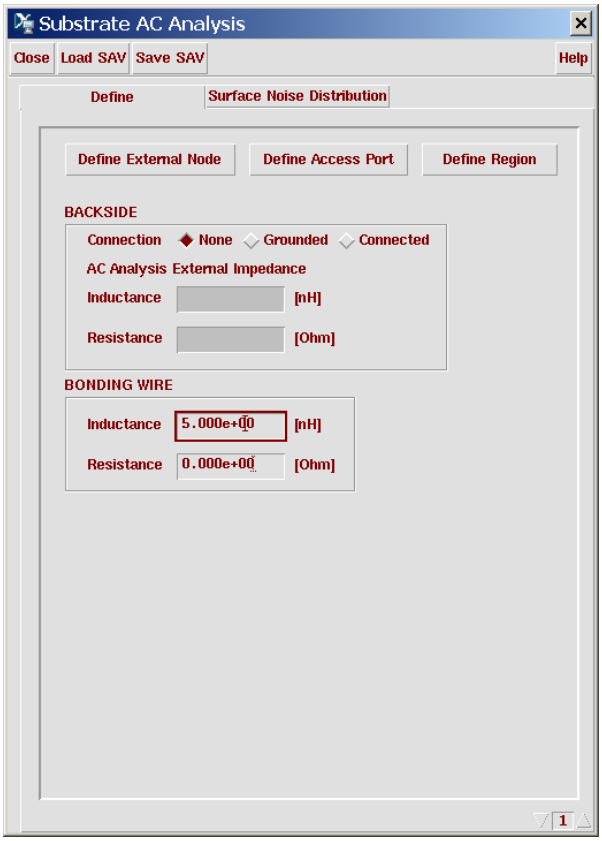
III.3 - Running Substrate AC Analysis

Note: For a complete description of the Substrate AC Analysis tool and the associated forms and options, please refer to the Assura Physical Verification User Guide (Product version 3.1.5_OA), Chapter 11.

Once you have obtained an `av_extracted` view with substrate parasitics, you can run the Substrate AC Analysis tool included in the Assura environment. This tool performs an AC analysis on the extracted circuit, and plots a surface noise distribution on the layout, which lets you identify the amount of noise reaching the different layout regions.

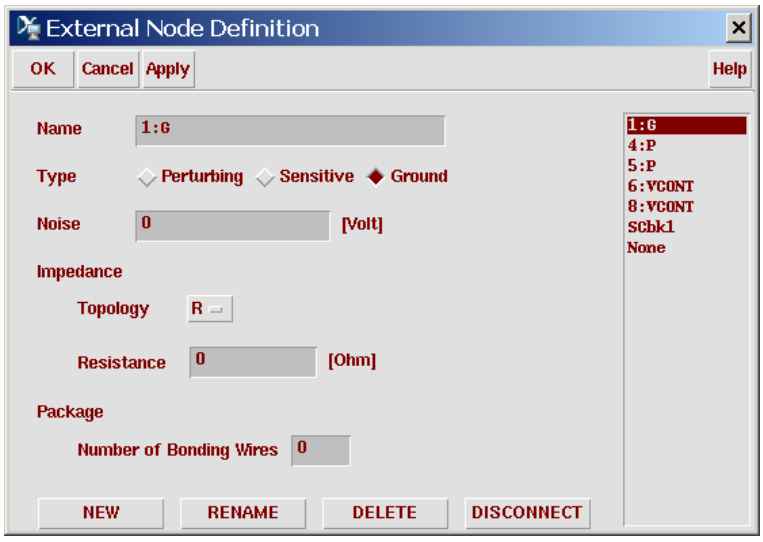
For example, the following figure shows the `av_extracted` view of a VCO layout (differential inductor not included, just its guard ring can be observed).

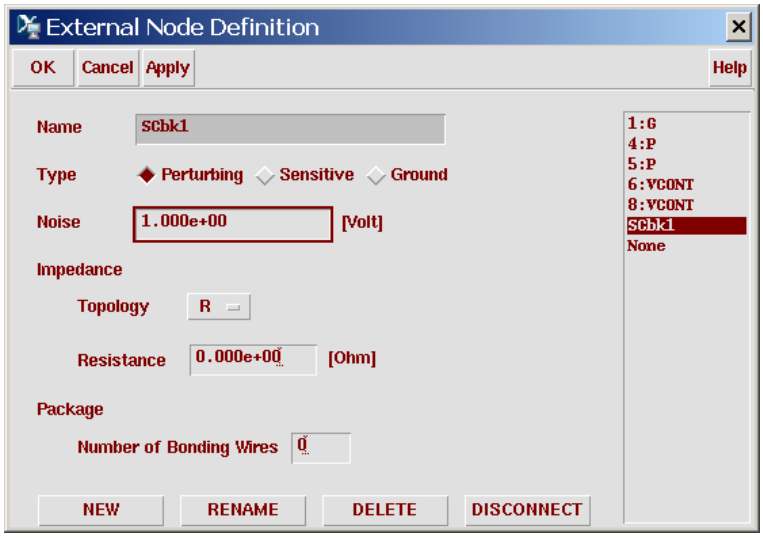
From this `av_extracted` view window, the `Substrate AC Analysis` can be called in the `Assura` menu. After loading the `av_extracted` SAV (Substrate Abstract View), the following form appears:



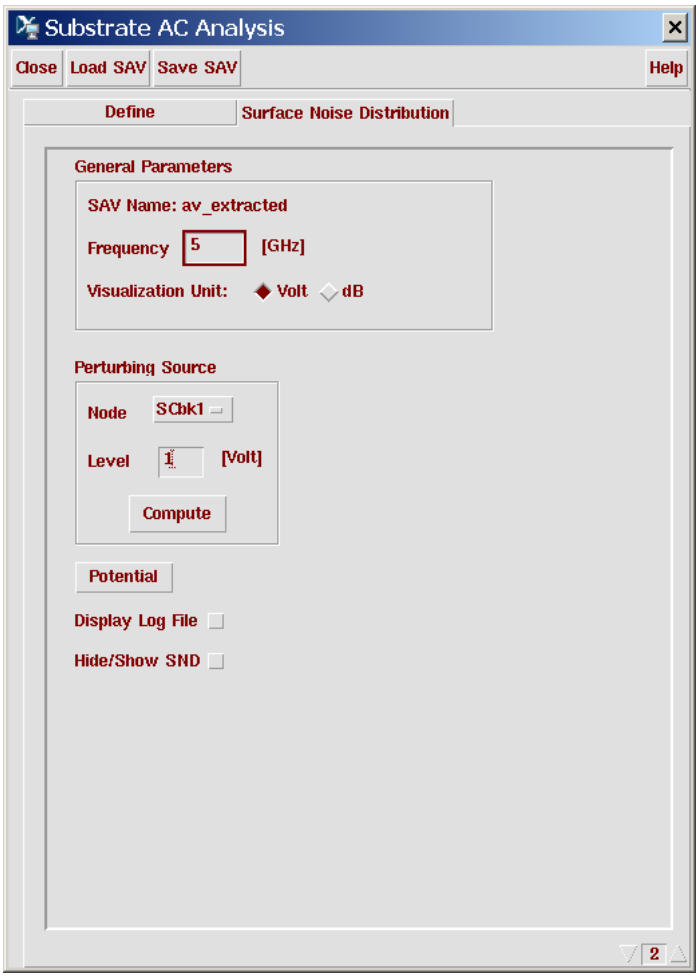
Inductance: You *must* specify a nominal inductance value for bonding wire and package pin parasitics. Circuit ground nodes will be connected to the ideal external ground through this inductance. Other nodes can be connected through these inductance, too (multiple bonding is allowed).

Once the Backside and Bonding Wire parasitics have been specified, you will click on the Define External Node box. Here, a list of external nodes appears. You should assign at least one of these nodes as Ground (inductive connection is assumed), and one of the nodes as Perturbing (you must specify the AC magnitude). Other nodes will be considered Sensitive, you can specify a load value for these Sensitive nodes, as well as bonding wire (inductance).





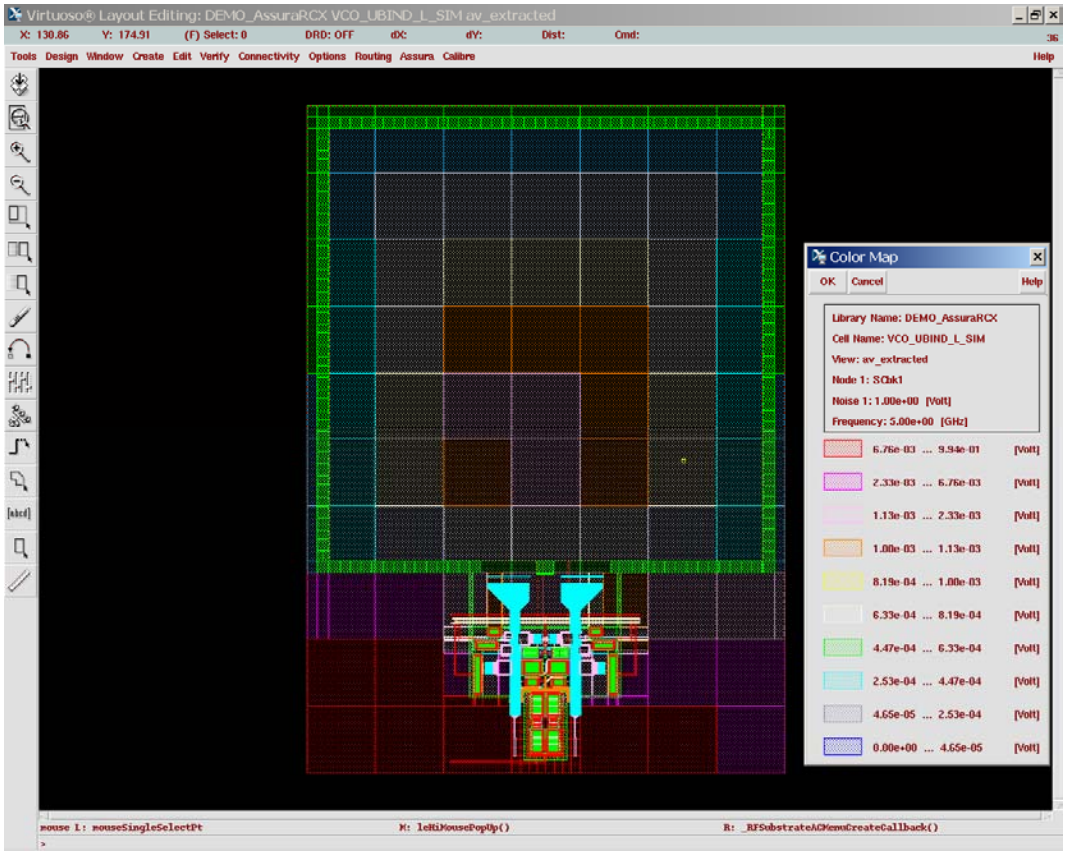
Once the Perturbing node is specified, click on the Surface Noise Distribution tab.

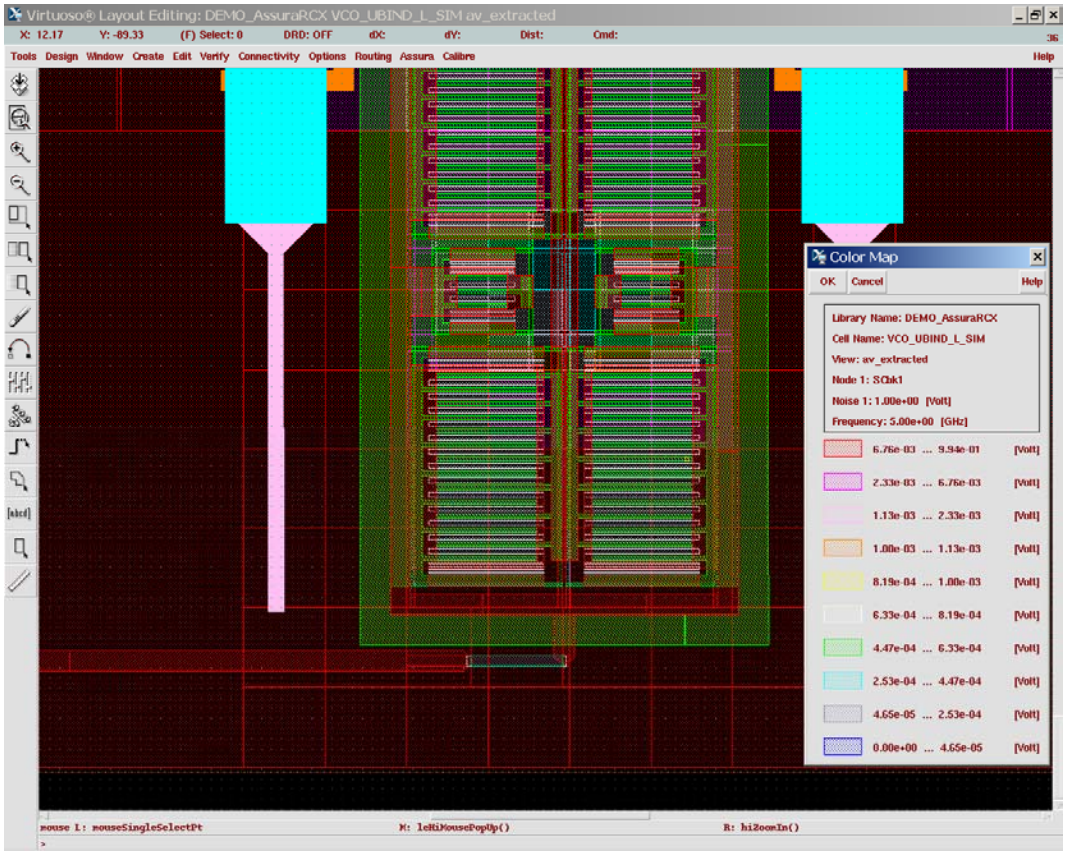


It is mandatory here to specify the frequency at which the AC analysis will be performed. The Perturbing Source values should be those specified in the External Node Definition form.

It is possible to define up to two different Perturbing Sources, and calculate their combines effect.

Once the forms have been completed, press the `Compute` button, the `av_extracted` view window will now show a surface noise map, and a color interpretation window specifies the relative noise levels corresponding to every color. In the results for the VCO circuit example, the reddish colors show the areas with highest noise levels, while the blue colors specify the most quiet regions.


Virtuoso® Layout Editing: DEMO_AssuraRCX VCO_UBIND_L_SIM av_extracted
X: 130.86 Y: 174.91 (F) Select: 0 DRD: OFF dX: dY: Dist: Cmd:
Tools Design Window Create Edit Verify Connectivity Options Routing Assura Calibre Help
Color Map
OK Cancel Help
Library Name: DEMO_AssuraRCX
Cell Name: VCO_UBIND_L_SIM
View: av_extracted
Node 1: SClk1
Noise 1: 1.00e+00 [Volt]
Frequency: 5.00e+00 [GHz]
6.76e-03 ... 9.94e-01 [Volt]
2.33e-03 ... 6.76e-03 [Volt]
1.13e-03 ... 2.33e-03 [Volt]
1.00e-03 ... 1.13e-03 [Volt]
8.19e-04 ... 1.00e-03 [Volt]
6.33e-04 ... 8.19e-04 [Volt]
4.47e-04 ... 6.33e-04 [Volt]
2.53e-04 ... 4.47e-04 [Volt]
4.65e-05 ... 2.53e-04 [Volt]
0.00e+00 ... 4.65e-05 [Volt]
mouse L: mouseSingleSelectPt M: leMiMousePopUp() R: _RFSubstrateACMenuCreateCallback()


Virtuoso® Layout Editing: DEMO_AssuraRCX VCO_UBIND_L_SIM av_extracted
X: 12.17 Y: -89.33 (F) Select: 0 DRD: OFF dX: dY: Dist: Cmd:
Tools Design Window Create Edit Verify Connectivity Options Routing Assura Calibre Help
Color Map
OK Cancel Help
Library Name: DEMO_AssuraRCX
Cell Name: VCO_UBIND_L_SIM
View: av_extracted
Node 1: SClk1
Noise 1: 1.00e+00 [Volt]
Frequency: 5.00e+00 [GHz]
6.76e-03 ... 9.94e-01 [Volt]
2.33e-03 ... 6.76e-03 [Volt]
1.13e-03 ... 2.33e-03 [Volt]
1.00e-03 ... 1.13e-03 [Volt]
8.19e-04 ... 1.00e-03 [Volt]
6.33e-04 ... 8.19e-04 [Volt]
4.47e-04 ... 6.33e-04 [Volt]
2.53e-04 ... 4.47e-04 [Volt]
4.65e-05 ... 2.53e-04 [Volt]
0.00e+00 ... 4.65e-05 [Volt]
mouse L: mouseSingleSelectPt M: leMiMousePopUp() R: hiZoomIn()

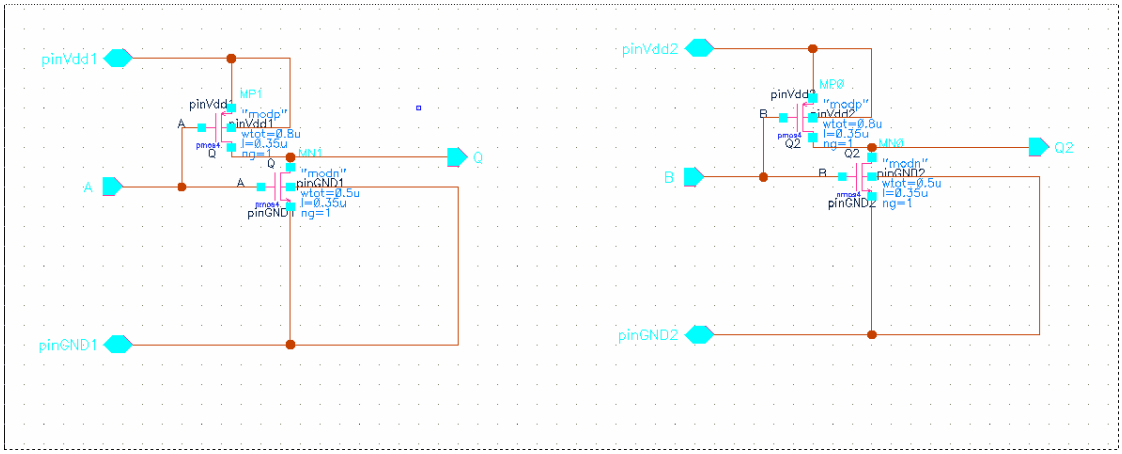
Appendix A. Examples

A.1 - Two inverters

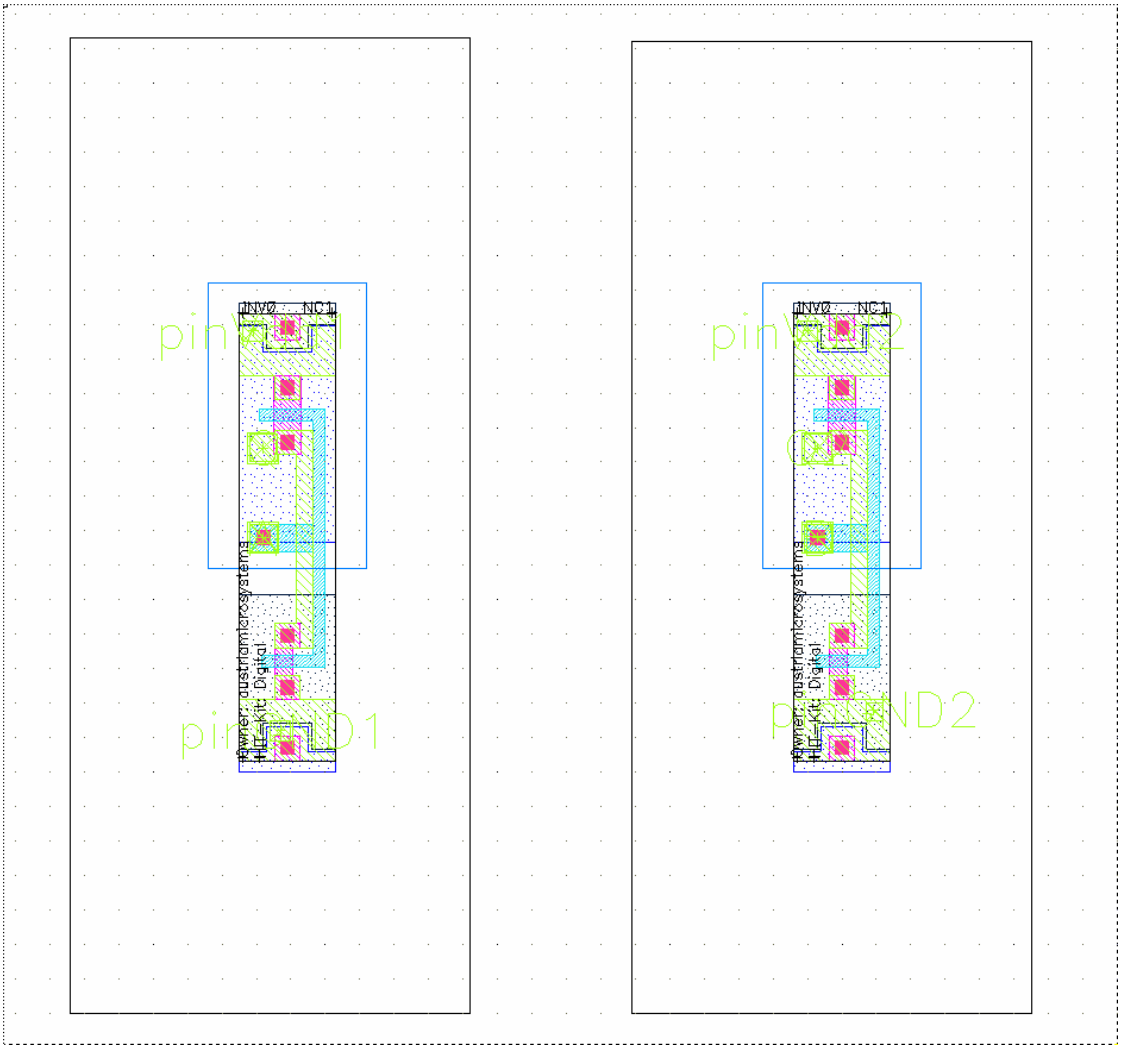
As an example to illustrate the files generated when the substrate parasitics are extracted, a simple circuit consisting of two inverters, connected to independent supply domain, is presented.

The example is found in the library "`AssuraTest`", cell "`TestInverters`".

The following picture shows the schematic of the circuit. The independent supply voltages, and the local connections of the substrate/well can be appreciated.



The layout of the circuit is shown next. Note the rectangles in SUBDEF layer that enclose each supply-voltage domain.



We run the LVS and run RCX extraction, with substrate (Assura RCX-HF), extract R Only, filter resistors smaller than 1 Ω.

Complete extraction takes a few seconds. After the run, the following substrate subcircuit (`substrate.subckt`) is stored under the `/AssuraTest/.sna/TestInverters/av_extracted` directory:

```
.SUBCKT SCav_extracted SCbk3 SCbk8 SCbk6 SCbk1 SCbk4 SCbk7
SCbk5 SCbk2

R1 33 SCbk3 29.3434
R2 77 SCbk3 19.3048
R3 39 SCbk8 1113.12
R4 83 SCbk8 1391.4
R5 56 SCbk6 15898.3
R6 60 SCbk6 2861.69
R7 95 SCbk6 1882.69
R8 99 SCbk6 15898.3
R9 62 SCbk1 206.4
R10 69 SCbk1 97.1294
R11 97 SCbk1 135.789
R12 110 SCbk1 63.9009
```

```
R13 175 SCbk4 11.6442
R14 184 SCbk7 618.4
R15 200 SCbk5 15898.3
R16 204 SCbk5 1135.59
R17 208 SCbk5 15898.3
R18 206 SCbk2 81.9048
R19 222 SCbk2 38.5434
R20 243 17 83.865
R21 244 17 12500.1
R22 245 17 14997.5
R23 244 18 12500.1
R24 269 18 83.865
R25 270 18 12500.1
R26 271 18 14997.5
...
R8577 4529 4528 2146.94
R8578 4530 4529 14395.4
R8579 4531 4530 112685
R8580 4532 4531 667705
R8581 4533 4532 3.83456e+06
C1 914 856 1.36044e-16
C2 914 873 2.6558e-16
C3 883 882 1.2054e-16
C4 885 884 1.37025e-16
...
C413 4249 4248 6.49175e-17
C414 4250 4249 3.32329e-17
C415 4261 4249 2.83789e-16
C416 4295 4249 1.50445e-16
.ENDS SCav_extracted
```

Note that the subcircuit has 8 external nodes (ports) named `SCbk1` ..`SCbk8`, that correspond to the bulk terminals of the four transistors, the two substrate taps, and the two N-well taps. Since no RC reduction was applied, the size of the generated netlist if remarkable even for such a simple circuit (4517 internal nodes, 8581 resistors, 416 capacitors for the well-substrate junctions)

We can generate a netlist for the complete circuit, including the substrate. Since the "`Import Netlist`" option was selected in the "`Substrate`" tab, the substrate subcircuit was included *flat* in the netlist. The following shows a section of the resulting netlist:

```
...
snaR26 (sna271 sna18) resistor r=14997.5
snaR25 (sna270 sna18) resistor r=12500.1
snaR24 (sna269 sna18) resistor r=83.865
snaR23 (sna244 sna18) resistor r=12500.1
snaR22 (sna245 sna17) resistor r=14997.5
snaR21 (sna244 sna17) resistor r=12500.1
snaR20 (sna243 sna17) resistor r=83.865
snaR19 (sna222 SCbk2) resistor r=38.5434
snaR18 (sna206 SCbk2) resistor r=81.9048
snaR17 (sna208 SCbk5) resistor r=15898.3
snaR16 (sna204 SCbk5) resistor r=1135.59
snaR15 (sna200 SCbk5) resistor r=15898.3
snaR14 (sna184 SCbk7) resistor r=618.4
snaR13 (sna175 SCbk4) resistor r=11.6442
snaR12 (sna110 SCbk1) resistor r=63.9009
snaR11 (sna97 SCbk1) resistor r=135.789
```

```
snaR10 (sna69 SCbk1) resistor r=97.1294
snaR9 (sna62 SCbk1) resistor r=206.4
snaR8 (sna99 SCbk6) resistor r=15898.3
snaR7 (sna95 SCbk6) resistor r=1882.69
snaR6 (sna60 SCbk6) resistor r=2861.69
snaR4 (sna83 SCbk8) resistor r=1391.4
snaR3 (sna39 SCbk8) resistor r=1113.12
snaR2 (sna77 SCbk3) resistor r=19.3048
snaR1 (sna33 SCbk3) resistor r=29.3434
snaR5 (sna56 SCbk6) resistor r=15898.3
avD859_2 (_3\:pinGND2 SCbk4) resistor r=1e-9 m=1 isnoisy=yes
avD859_1 (_3\:pinGND1 SCbk3) resistor r=1e-9 m=1 isnoisy=yes
avD865_2 (_3\:pinVdd2 SCbk2) resistor r=1e-9 m=1 isnoisy=yes
avD865_1 (_3\:pinVdd1 SCbk1) resistor r=1e-9 m=1 isnoisy=yes
MN1 (_2\:Q _5\:A _2\:pinGND1 SCbk8) modn w=5e-07 l=3.5e-07
as=0.6025p \
        ad=0.6025p ps=2.75u pd=2.75u nrd=7.200e-01
nrs=7.200e-01 ng=1
MN0 (_2\:Q2 _5\:B _5\:pinGND2 SCbk7) modn w=5e-07 l=3.5e-07
as=0.6025p \
        ad=0.6025p ps=2.75u pd=2.75u nrd=7.200e-01
nrs=7.200e-01 ng=1
MP1 (_2\:pinVdd1 _3\:A _3\:Q SCbk6) modp w=8e-07 l=3.5e-07
as=0.78p \
        ad=0.76p ps=2.75u pd=2.7u nrd=5.349e-01 nrs=5.662e-01
ng=1
MP0 (_2\:pinVdd2 _3\:B _3\:Q2 SCbk5) modp w=8e-07 l=3.5e-07
as=0.78p \
        ad=0.76p ps=2.75u pd=2.7u nrd=5.349e-01 nrs=5.662e-01
ng=1
re3 (_2\:Q Q) resistor r=30.8206
re6 (_3\:Q Q) resistor r=60
re7 (_2\:pinVdd1 pinVdd1) resistor r=60.1245
re10 (_3\:pinVdd1 pinVdd1) resistor r=30
re11 (_2\:pinGND1 pinGND1) resistor r=30.0796
re13 (pinGND1 _3\:pinGND1) resistor r=60.0125
re16 (_2\:Q2 Q2) resistor r=30.8206
re19 (_3\:Q2 Q2) resistor r=60
re20 (_2\:pinVdd2 pinVdd2) resistor r=60.1245
re23 (_3\:pinVdd2 pinVdd2) resistor r=30
re26 (pinGND2 _5\:pinGND2) resistor r=30.1245
re27 (_3\:pinGND2 pinGND2) resistor r=60
rg1 (_3\:A _4\:A) resistor r=98.4653
rg2 (_4\:A A) resistor r=16.5
rg3 (_4\:A _5\:A) resistor r=101.323
rg4 (_3\:B _4\:B) resistor r=98.4653
rg5 (_4\:B B) resistor r=16.5
rg6 (_4\:B _5\:B) resistor r=101.323
```

Note that the bulk node of the transistors are connected to ports `Sbck5 .. Sbck8`. The taps are connected to the supply nodes through the resistors `acD859_2`, `avD859_1`, `avD865_2` and `avD865_1`. These resistors are the result of the extraction of the `short_ptap` and `short_ntap` devices. Note that the connection to the supply nodes is correctly made. Connectivity from the external pins to the substrate mesh can easily be traced in this example. Note that the two supply domains (1 and 2) are independent since there is no circuit element connecting nodes of both domains. Connection between domains is done only through the substrate subcircuit.

It is interesting to compare the netlist shown above, with the netlist obtained without substrate extraction:

```
avD859_2 (_3\:pinGND2 _3\:pinGND2) resistor r=1e-9 m=1 isnoisy=yes
avD859_1 (_3\:pinGND1 _3\:pinGND1) resistor r=1e-9 m=1 isnoisy=yes
avD865_2 (_3\:pinVdd2 _3\:pinVdd2) resistor r=1e-9 m=1 isnoisy=yes
avD865_1 (_3\:pinVdd1 _3\:pinVdd1) resistor r=1e-9 m=1 isnoisy=yes
MN1 (_2\:Q _5\:A _2\:pinGND1 _3\:pinGND1) modn w=5e-07 l=3.5e-07 \
        as=0.6025p ad=0.6025p ps=2.75u pd=2.75u nrd=7.200e-01 \
        nrs=7.200e-01 ng=1
MN0 (_2\:Q2 _5\:B _5\:pinGND2 _3\:pinGND2) modn w=5e-07 l=3.5e-07
\
        as=0.6025p ad=0.6025p ps=2.75u pd=2.75u nrd=7.200e-01 \
        nrs=7.200e-01 ng=1
MP1 (_2\:pinVdd1 _3\:A _3\:Q _3\:pinVdd1) modp w=8e-07 l=3.5e-07
as=0.78p \
        ad=0.76p ps=2.75u pd=2.7u nrd=5.349e-01 nrs=5.662e-01 ng=1
MP0 (_2\:pinVdd2 _3\:B _3\:Q2 _3\:pinVdd2) modp w=8e-07 l=3.5e-07
as=0.78p \
        ad=0.76p ps=2.75u pd=2.7u nrd=5.349e-01 nrs=5.662e-01 ng=1
re3 (_2\:Q Q) resistor r=30.8206
re6 (_3\:Q Q) resistor r=60
re7 (_2\:pinVdd1 pinVdd1) resistor r=60.1245
re10 (_3\:pinVdd1 pinVdd1) resistor r=30
re11 (_2\:pinGND1 pinGND1) resistor r=30.0796
re13 (pinGND1 _3\:pinGND1) resistor r=60.0125
re16 (_2\:Q2 Q2) resistor r=30.8206
re19 (_3\:Q2 Q2) resistor r=60
re20 (_2\:pinVdd2 pinVdd2) resistor r=60.1245
re23 (_3\:pinVdd2 pinVdd2) resistor r=30
re26 (pinGND2 _5\:pinGND2) resistor r=30.1245
re27 (_3\:pinGND2 pinGND2) resistor r=60
rg1 (_3\:A _4\:A) resistor r=98.4653
rg2 (_4\:A A) resistor r=16.5
rg3 (_4\:A _5\:A) resistor r=101.323
rg4 (_3\:B _4\:B) resistor r=98.4653
rg5 (_4\:B B) resistor r=16.5
rg6 (_4\:B _5\:B) resistor r=101.323
```

Note that the connections of the bulk nodes of the transistors is properly made. The `acD859_2`, `avD859_1`, `avD865_2` and `avD865_1` resistors persist, but their nodes have been shorted to the corresponding supply voltage.

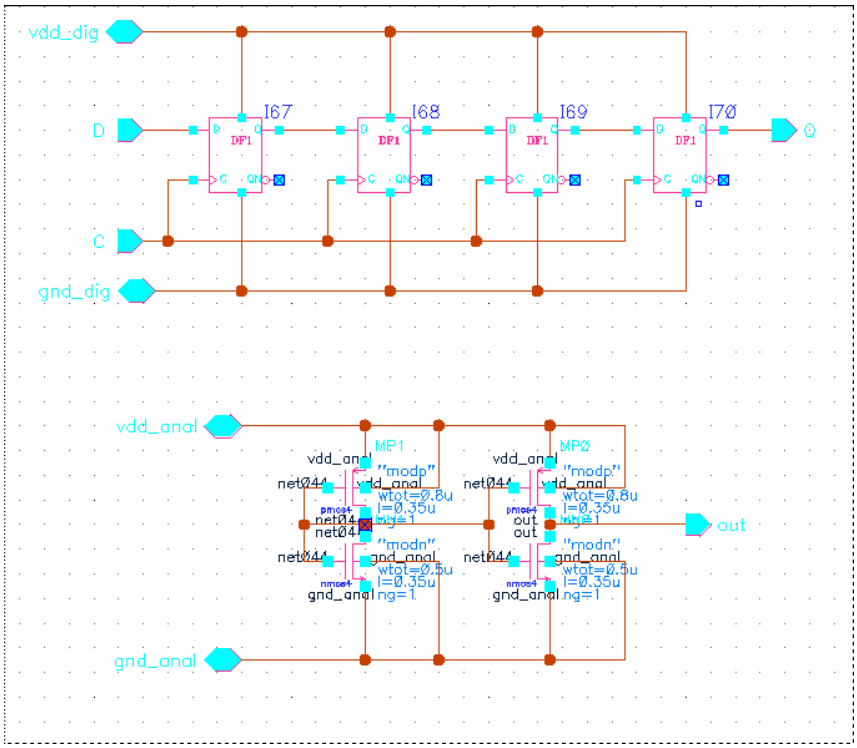
A.2 - Shift-register + amplifier

We show now a more complex circuit, which we will simulate and compare the simulation results with and without substrate extraction. The circuit consists of an 4-stage shift register as a noisy circuit, and a simple push-pull amplifier biased with a voltage divider.

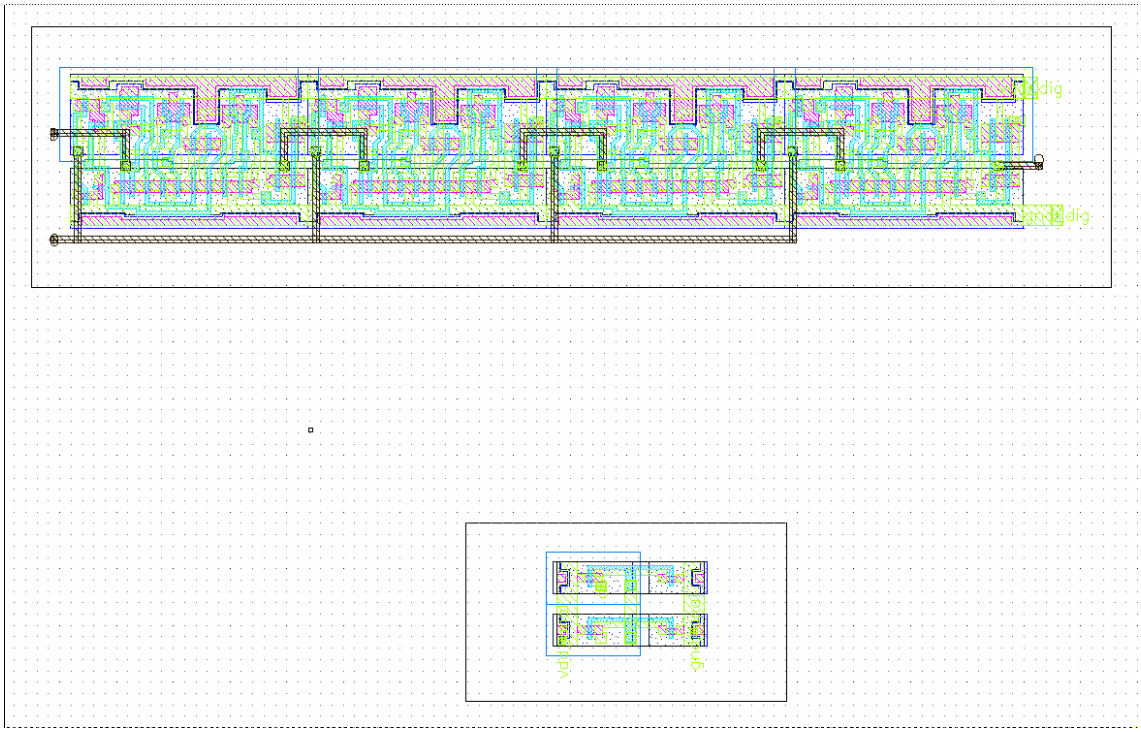
The example is found in the library "`AssuraTest`", cells "`TestDF4`".

The following picture shows the schematic of the circuit. Note that DFF symbols (and schematics) have been modified in order to eliminate the global nodes and replace them by

externally accessible nodes. This might not be necessary if on-chip digital and analog global nodes are defined.



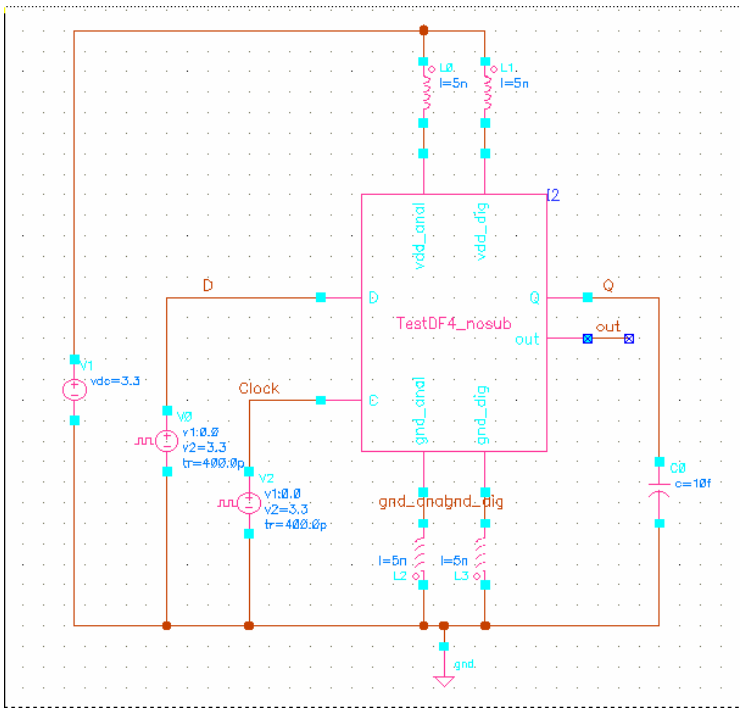
The layout of the circuit is shown next. Note the rectangles in SUBDEF layer that enclose each supply-voltage domain.



We run the LVS and run RCX extraction, first without substrate (Assura RCX-HF), extract RC, filter resistors smaller than 1 Ω and capacitors smaller than 10 fF.

Extraction takes a few seconds.

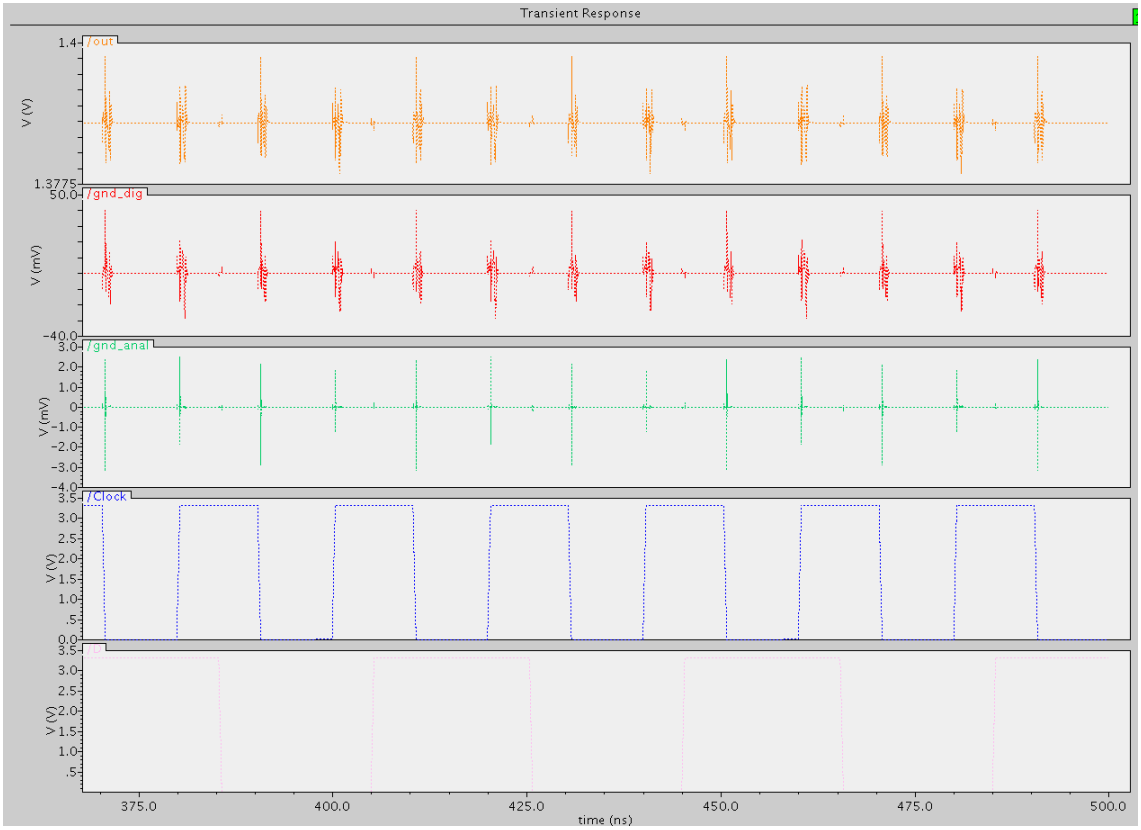
After the parasitics of the `TestDF4` cell have been extracted, a schematic cell called `TestSim_DF` is created. This schematic contains the simulation set-up for the circuit. Note that 5 nH inductances are connected to the different supply nodes, in order to provide a basic model of the package parasitics.



The following shows the circuit inventory of the generated netlist:

```
Circuit inventory:
                nodes 380
            equations 619
              bsim3v3 108
            capacitor 352
             inductor 4
             quantity 9
             resistor 371
              vsource 3
```

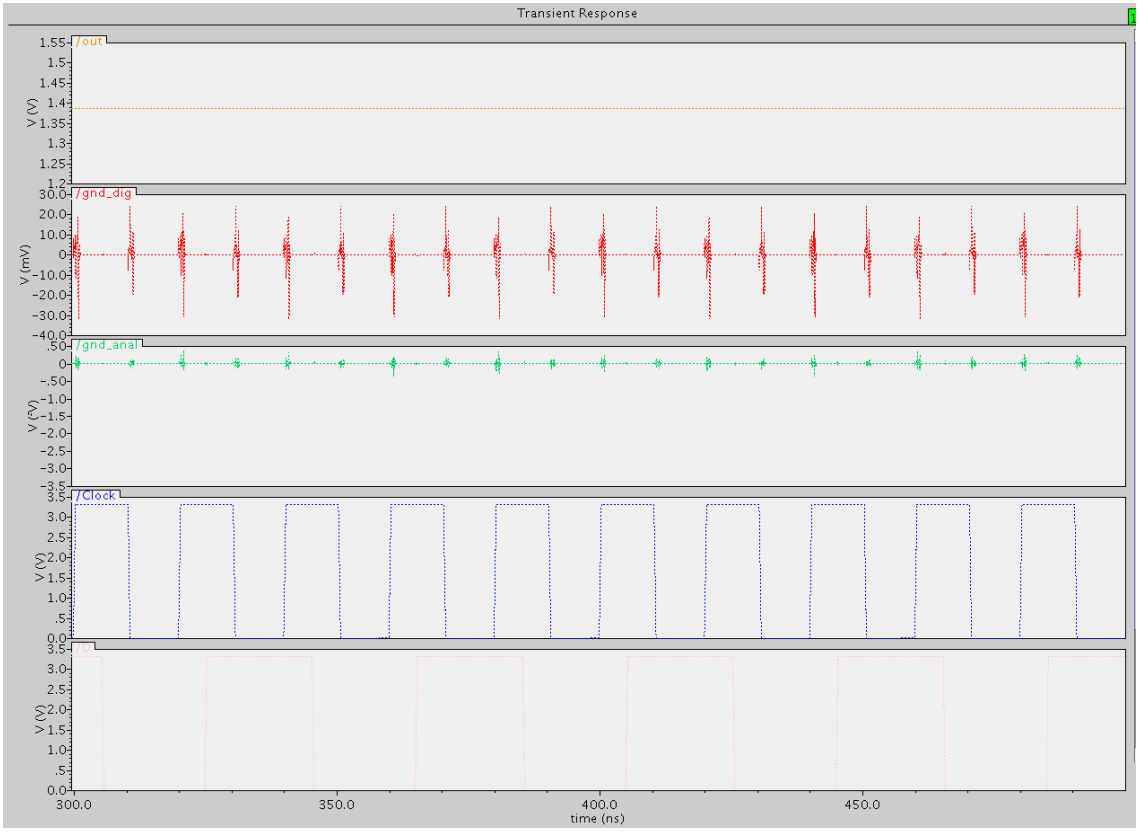
Since no substrate has been extracted now, the digital and analog circuits in the `TestDF4` cell should be completely independent, and therefore the analog nodes should not be affected by the switching noise generated by the digital circuit. A transient analysis is run in just a few seconds, and the results obtained are shown next:



It can be seen that, contrary to what is expected, the analog nodes `gnd_anal` and `out` are both affected by the digital noise. In particular the noise at the `out` node has an amplitude of about 10 mVpp.

The reason for this coupling is that the parasitic capacitances extracted by RCX are all referenced to the same node, which provides a coupling path between the digital and analog subcircuits. By default, the Reference Node (see Extraction tab in section III.2.2) is set to `dummy`. The simulator chooses a node (typically the gnd node with the most number of elements connected to it) as this dummy node. In our case, it was `gnd_dig`, which provoked that all the parasitic capacitances in the analog subcircuit were connected to the digital ground, thus the noise coupling.

The former problem can be circumvented by referencing the capacitances to an ideal gnd node. Thus, the loading effect of the parasitic capacitors is accounted for, but since the common node is quiet, no noise is transferred from the noisy to the quiet subcircuits. We tried to do this by selecting `gnd!` as the Reference Node, but since this `gnd!` node is not present in the `TestDF4` cell, the extractor again selected `gnd_dig` as the reference node for the capacitors. Therefore, the solution was to include an unconnected `gnd!` node both in the layout (a small piece of top metal) and schematics (since this is a global node, it will become connected to the ideal `gnd!` node of the `TestSim_DF` schematic). The following figure shows the simulation results with this set-up, where the `out` node is now quiet (note that the amplitude of `gnd_anal` is mere numerical noise).



Once we have checked that no noise is coupled to the sensitive `out` node if no substrate is extracted, let's now repeat the extraction including the substrate parasitics.

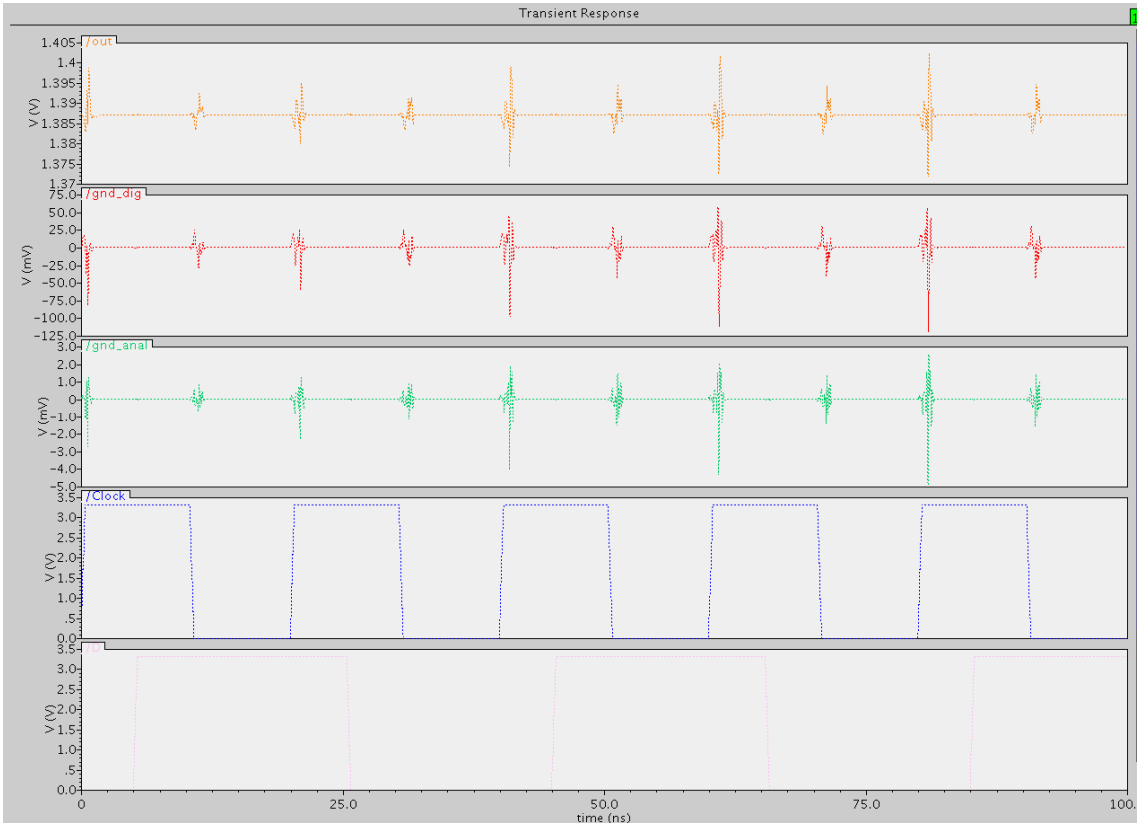
Complete extraction now takes about 20 minutes. The size of the substrate subcircuit is now huge, even for such a relatively small circuit. The number of external ports is 124, while there are 59,651 internal nodes. The number of resistors is 119.809, and the number of capacitors 9.544.

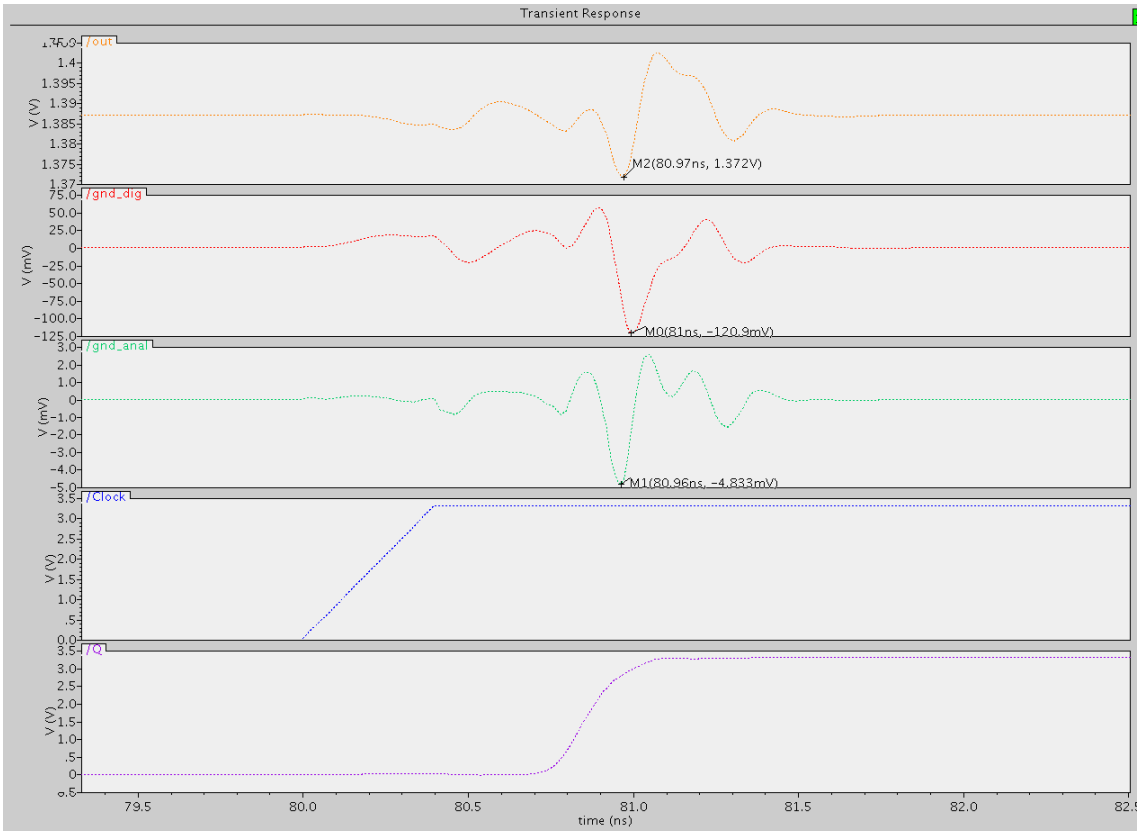
```
Number of external nodes:                    124
Number of original internal nodes:           59651
Number of internal nodes after reduction:    0
Number of original resistors:                119809
Number of resistors after reduction:         119809
Number of original capacitors:               9544
Number of capacitors after reduction:        9544
```

This new extraction is run with the same simulation set-up as before. The circuit inventory of the generated netlist is the following:

```
Circuit inventory:
              nodes 60155
          equations 60394
            bsim3v3 108
          capacitor 10200
           inductor 4
           quantity 9
           resistor 120180
            vsource 3
```

Simulation takes several hours, even for such a relatively small circuit. The following plot shows the waveforms obtained.



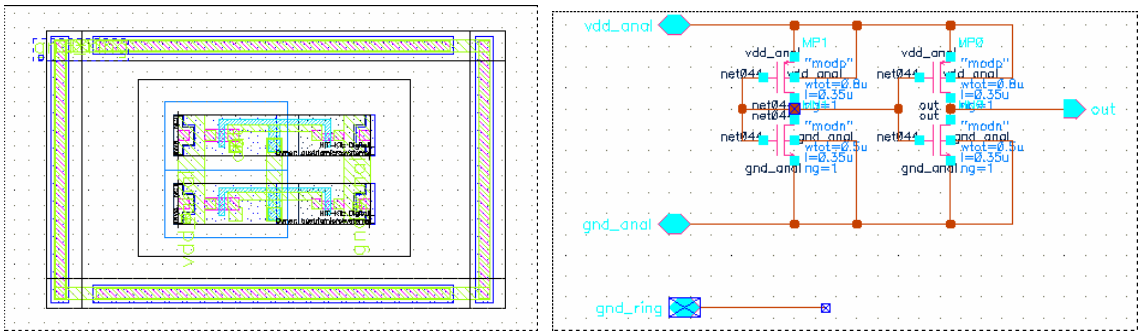


```
out: 1,402-1,372 = 30 mVpp
gnd_dig: 55,89m-(-120,9m)=176.79 mVpp
gnd_anal: 2,555mV-(-4,833mV)=7,388 mVpp
```

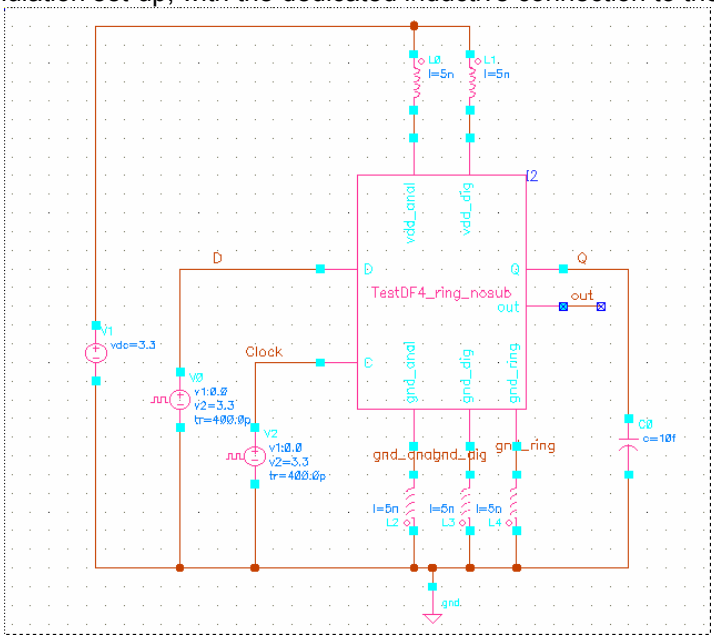
Es expected, the new extraction is now able to predict the substrate coupling to the sensitive node. Coupling is larger to the sensitive output node than to the `gnd_anal` node.

The former circuit has now been modified by adding a guard ring around the sensitive circuit. This ring consists of P+ diffusions grounded with a dedicated pin.

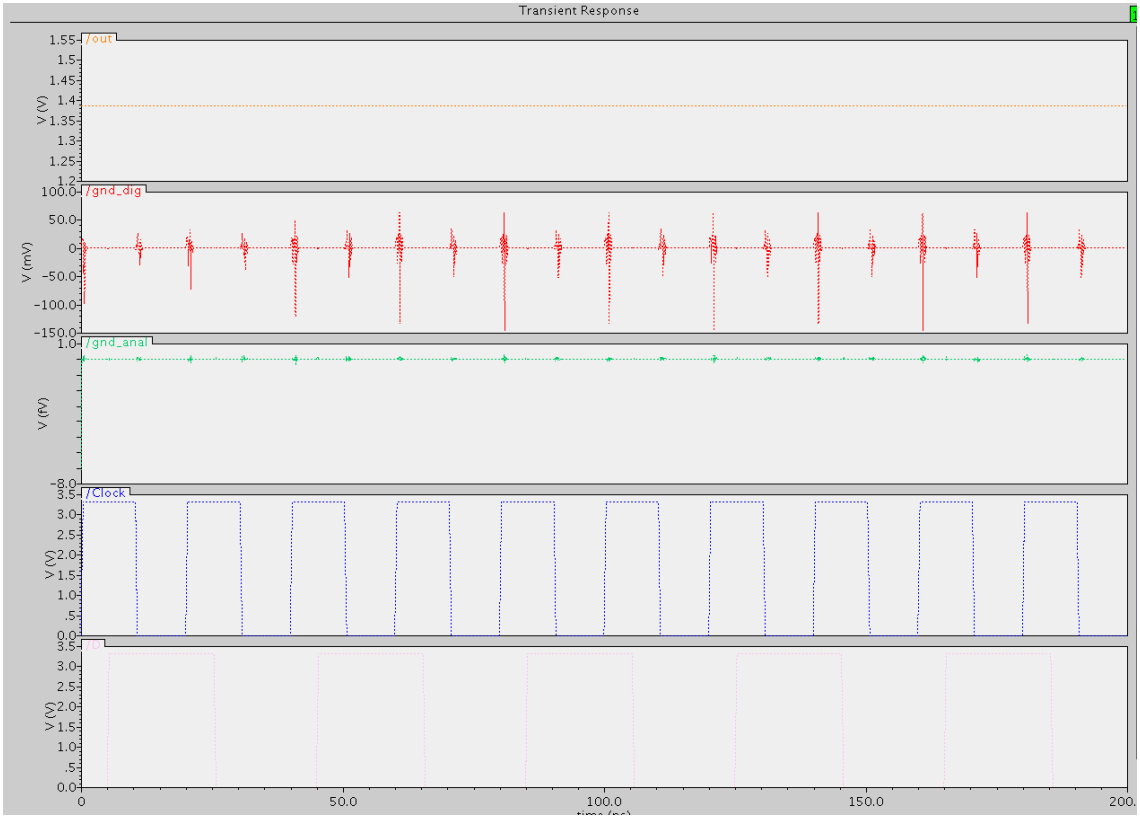
The following picture shows the modifications added to the analog section, both in the layout and schematic views.



And this is the simulation set-up, with the dedicated inductive connection to the guard ring:

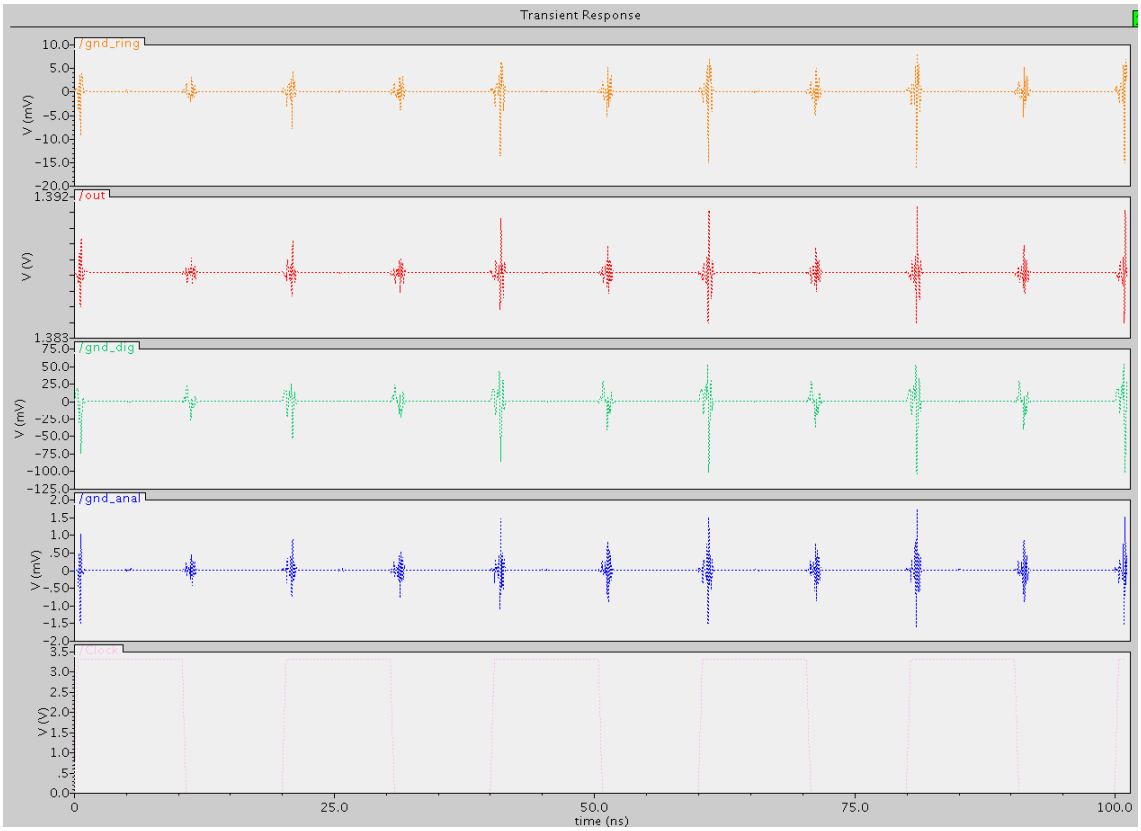


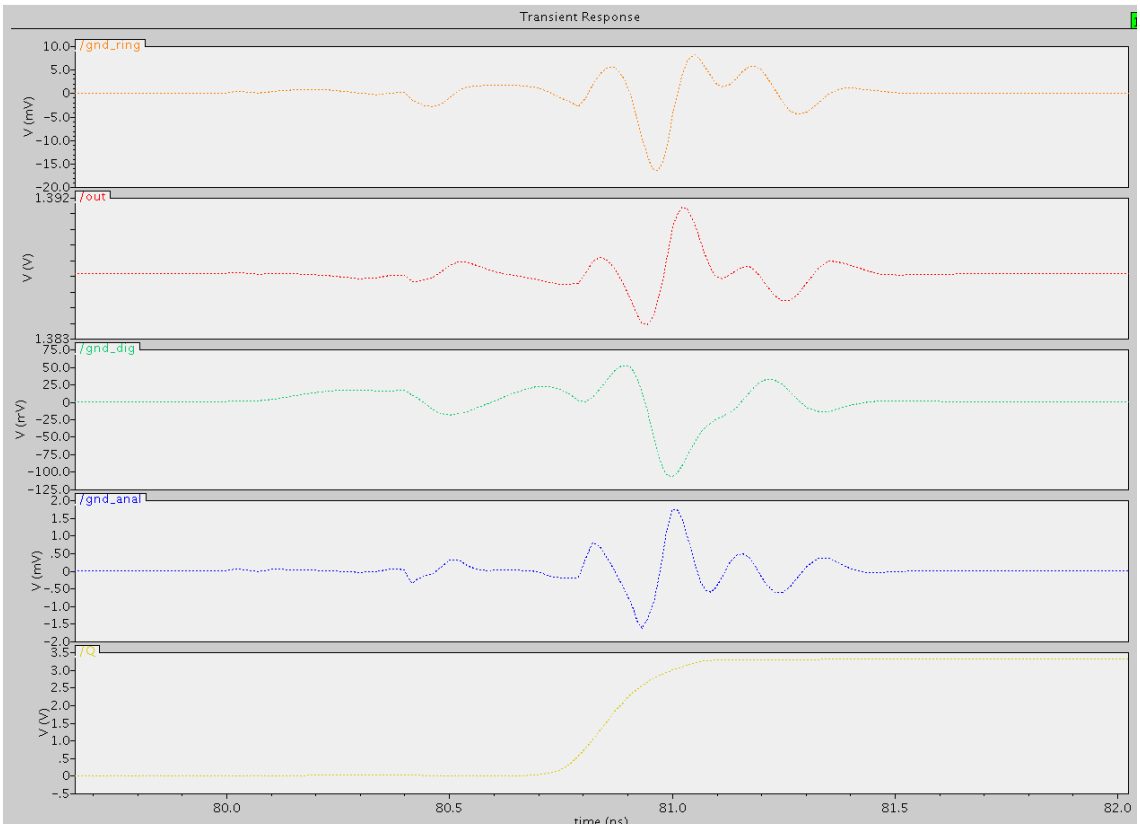
As expected, no noise is coupled to the sensitive `out` node if no substrate is extracted:



Extraction with substrate takes again about 20 minutes and the simulation takes several hours. Note that the ring has also been connected to `gnd` through a 5 nH inductor.

The following shows the result of the simulation:





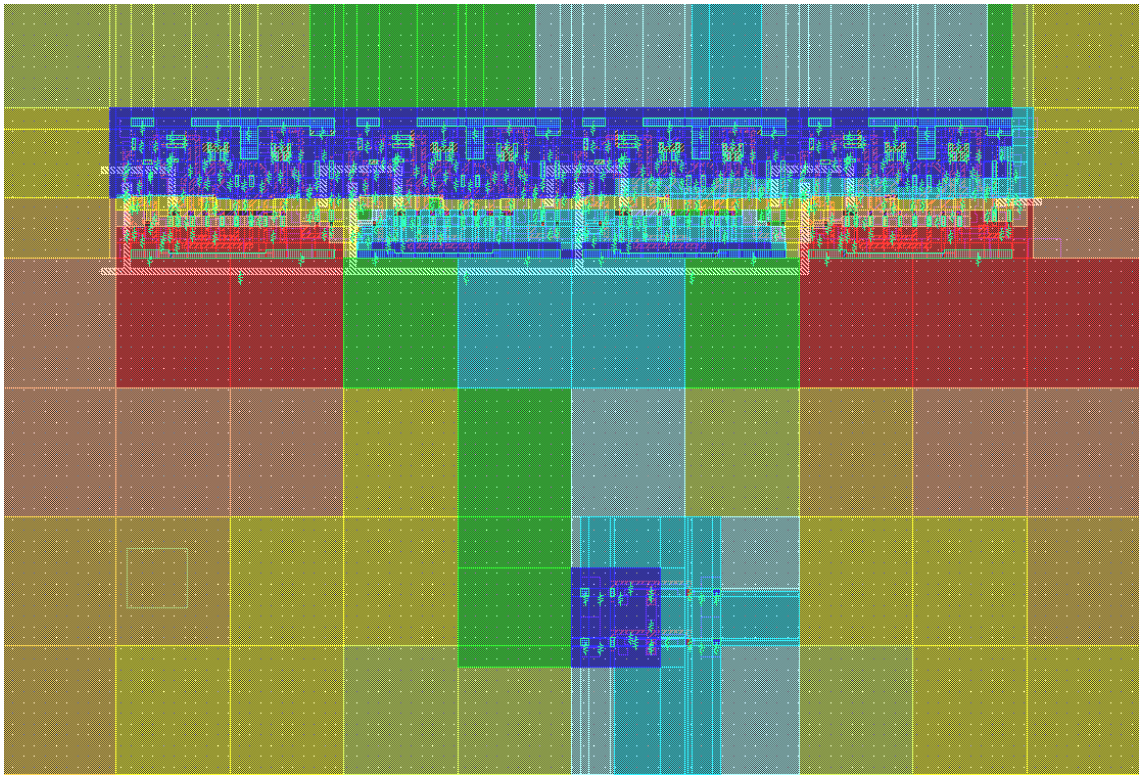
```
gnd_ring: 8,079m-(-16,52m)=24,599 mVpp
out: 1,391-1,384=7 mVpp
gnd_dig: 52m-(-106,9m)=158,9 mVpp
gnd_anal: 1,729mV-(-1,618mV)=3,347 mVpp
```

The simulation with the extracted parasitics show how the ring is able to reduce the noise at the sensitive node to less than one fourth of the original layout. Note that the `gnd_anal` node is now extremely quiet, showing how currents have chosen `gnd_ring` as the return path, letting the ring inside quiet.

Once the extractions are available, the AC_analysis utility can be run to generate noise colour maps of the noise distribution.

From the netlist of the `TestSim_DF` circuit, we identified `SCbk13`, `SCbk14`, `SCbk15`, `SCbk16` as the substrate nodes connected to `gnd_dig`, therefore we defined those nodes as noisy. Also we identified `SCbk11` and `SCbk12` as the substrate nodes connected to `gnd_anal`, and `SCbk1` and `SCbk2` as the substrate nodes connected to `vdd_anal`, therefore, we defined those nodes as ground. We defined an inductance of 5 nH. Unfortunately, AC_analysis form only allows two nodes to be defined as simultaneous noise generators, thus we chose `SCbk13` and `SCbk16` (press Combine to add the contributions of both sources).

The following shows the plot generated. Note that the noisy sources are placed at the gnd nodes of the extreme flip-flops. The effect of the analog ground connections is evident, particularly inside the n-well.



For comparison, we repeated the same analysis with the `TestSim_DF_ring` circuit. Nodes connected to gnd_dig are now `SCbk121`, `SCbk122`, `SCbk123`, `SCbk124`, nodes connected to the `gnd_anal` are `SCbk119`, `SCbk120`, nodes connected to `vdd_anal` are `SCbk109`, `SCbk110`, and nodes connected to the guar ring are `SCbk125`, `SCbk126`, `SCbk127`, `SCbk128`.

The following is the color map obtained. We can see that for the same noise sources, the ring inside is now significantly more quiet, thus demonstrating the efficacy of the ring, and the reason of the low noise levels for the `out` and `gnd_anal` nodes ontained in the simulation.